



# MODELADO 3D DEL CAMPUS UC3M DE LEGANÉS PARA SU INTEGRACIÓN EN UN PROTOTIPO DE VIDEOJUEGO FPS



UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR  
GRADO EN INGENIERÍA INFORMÁTICA

**Autor:** *Miguel Bohada Arranz*  
**Tutor:** *Gustavo Recio Isasi*  
**Co-Tutor:** *Alejandro Baldominos Gómez*  
**Fecha:** *Junio 2015*

# Agradecimientos

---

*A mis padres y a mi hermano, por apoyarme siempre y hacer posible que haya llegado hasta aquí.*

*A Cristina, por su ayuda incondicional durante este largo trayecto.*

*A mi tutor, Gustavo, por darme la oportunidad de trabajar en este proyecto y animarme a seguir hasta el final.*

*Y a todos aquellos que han estado ahí de una forma u otra en cada etapa de este proyecto.*

GRACIAS.

# Resumen

---

En este proyecto se muestra el proceso de creación de una visita virtual al edificio Agustín de Betancourt del campus de Leganés de la Universidad Carlos III de Madrid utilizando las técnicas de creación de un videojuego, de tal manera que el usuario pueda moverse e interactuar libremente por todo el edificio sin ninguna restricción. Para poder llegar a un mayor número de personas y que cualquier usuario sea capaz de tener acceso a esta visita virtual, se ha optado por basar el método de interacción en el movimiento mediante teclado y ratón, dado que son los dos periféricos básicos que todo ordenador posee, evitando que se necesiten adquirir nuevos componentes.

La decisión de crear una visita virtual interactiva viene marcada por la situación actual en la que se encuentra el desarrollo de las visitas virtuales estudiadas (centradas en universidades tanto nacionales como internacionales), en las que se ha comprobado que al usuario no se le permite examinar con libertad el entorno en el que está interesado, y tiene que conformarse con ver fotografías panorámicas o vídeos de lugares muy concretos.

Para poder realizar este proyecto ha sido necesario analizar las herramientas disponibles en la actualidad para modelar entornos tridimensionales, los motores de juego que permiten implementar los modelos realizados anteriormente y los múltiples sistemas de interacción.

Tras hacer un análisis de los requisitos y necesidades del proyecto, se tuvieron que buscar planos, documentación técnica sobre edificación y múltiples muestras fotográficas que permitiesen crear una experiencia realista.

Para llevar a cabo este proyecto se dividió el trabajo en dos partes: en la primera se diseñó el edificio Betancourt con Blender, dotando a los modelos de realismo mediante técnicas de texturizado; mientras que en la segunda, con Unity 3D, se tuvo que trabajar con iluminación, colisiones y sistemas de interacción, para que al implementar los modelos creados anteriormente en el escenario, el usuario pudiese moverse libremente y sentir una experiencia inmersiva.

De esta manera, se pretende conseguir que los usuarios puedan conocer completamente el edificio Betancourt sin desplazarse de casa, experimentando una sensación de inmersión que no se ha encontrado en otras visitas virtuales.

## Índice de Contenidos

1. Introducción y objetivos.....	10
2. Estado del arte .....	13
2.1. Visitas virtuales .....	13
2.2. Industria de los videojuegos.....	17
2.3. Diseño 3D .....	24
2.4. Interacción.....	34
2.5. Conclusiones.....	36
3. Definición y desarrollo del problema.....	38
3.1. Definición del problema .....	38
3.2. Desarrollo del problema.....	39
3.2.1. Análisis del problema .....	39
3.2.2. Diseño de la solución.....	41
3.2.3. Gestión de datos .....	49
4. Implementación .....	53
4.1. Modelado .....	53
4.1.1. Estructura del edificio Agustín de Betancourt .....	54
4.1.2. Decoración interior .....	62
4.1.3. Texturizado de modelos .....	66
4.2. Motor de juego.....	74
4.2.1. Importación de modelos .....	74
4.2.2. Uso de recursos externos.....	76
4.2.3. Iluminación.....	79
4.2.4. Interacción.....	80
4.2.5. Animaciones .....	84
4.2.6. Menús.....	85
4.3. Resultado final.....	88
5. Pruebas.....	92
5.1. Pruebas realizadas y resultados obtenidos.....	92
5.2 Análisis de los resultados .....	92
6. Planificación y presupuesto .....	95
6.1. Planificación .....	95
6.1.1. Estimada.....	96



6.1.2. Real .....	97
6.1.3. Metodología .....	98
6.2. Presupuesto .....	98
6.2.1. Recursos Humanos .....	99
6.2.2. Recursos materiales .....	99
6.2.3. Costes totales .....	100
7. Marco regulador y entorno socio-económico.....	101
7.1. Marco regulador.....	101
7.2. Entorno socio-económico .....	102
8. Conclusiones y trabajos futuros .....	104
9. Bibliografía .....	108
Anexo I: Acrónimos .....	110
Anexo II: Glosario de términos.....	111
Anexo III: Requisitos del sistema.....	112
Requisitos software funcionales .....	113
Requisitos software no funcionales .....	115
Anexo IV: Casos de uso .....	116
Anexo V: Scripts.....	121
Anexo VI: Manual de usuario .....	126

## Índice de tablas

<i>Tabla 1: Comparativa programas de modelado .....</i>	<i>33</i>
<i>Tabla 2: Comparativa motores de videojuegos.....</i>	<i>33</i>
<i>Tabla 3: Resumen de requisitos funcionales .....</i>	<i>40</i>
<i>Tabla 4: Resumen de requisitos no funcionales .....</i>	<i>40</i>
<i>Tabla 5: Resumen de casos de uso .....</i>	<i>41</i>
<i>Tabla 6: Relación entre Requisitos del sistema y Casos de uso .....</i>	<i>41</i>
<i>Tabla 7: Cuestionario de evaluación de pruebas.....</i>	<i>93</i>
<i>Tabla 8: Resultados de las pruebas .....</i>	<i>94</i>
<i>Tabla 9: Estimación del presupuesto de recursos humanos.....</i>	<i>99</i>
<i>Tabla 10: Estimación presupuesto para hardware .....</i>	<i>99</i>
<i>Tabla 11: Estimación presupuesto para software.....</i>	<i>99</i>
<i>Tabla 12: Estimación total del presupuesto .....</i>	<i>100</i>
<i>Tabla 13: Plantilla para requisitos funcionales y requisitos no funcionales.....</i>	<i>112</i>
<i>Tabla 14: RSF-01 - Menú Principal .....</i>	<i>113</i>
<i>Tabla 15: RSF-02 - Sistema de juego FPS.....</i>	<i>113</i>
<i>Tabla 16: RSF-03 - Escenario .....</i>	<i>113</i>
<i>Tabla 17: RSF-04 - Interacción.....</i>	<i>113</i>
<i>Tabla 18: RSF-05 - Menú Pausa.....</i>	<i>114</i>
<i>Tabla 19: RSF-06 - Calidad Gráfica .....</i>	<i>114</i>
<i>Tabla 20: RSF-07 - Salir Aplicación .....</i>	<i>114</i>
<i>Tabla 21: RSNF-01 - Multiplataforma .....</i>	<i>115</i>
<i>Tabla 22: RSNF-02 - Compatibilidad con resoluciones .....</i>	<i>115</i>
<i>Tabla 23: RSNF-03 - Fichero ejecutable.....</i>	<i>115</i>
<i>Tabla 24: Plantilla para casos de uso .....</i>	<i>116</i>
<i>Tabla 25: CU-01 - Visita Virtual .....</i>	<i>117</i>
<i>Tabla 26: CU-02 - Salir desde Menú Principal .....</i>	<i>117</i>
<i>Tabla 27: CU-03 - Mover Cámara.....</i>	<i>118</i>
<i>Tabla 28: CU-04 - Mover Personaje.....</i>	<i>118</i>
<i>Tabla 29: CU-05 - Abrir/Cerrar Puerta.....</i>	<i>119</i>
<i>Tabla 30: CU-06 - Ir a Menú Principal .....</i>	<i>119</i>
<i>Tabla 31: CU-07 - Cambiar Calidad Gráfica.....</i>	<i>120</i>
<i>Tabla 32: CU-08 - Salir desde Menú Pausa .....</i>	<i>120</i>

## Índice de ilustraciones

<i>Ilustración 1: Universidad Carlos III de Madrid, Campus de Leganés</i>	14
<i>Ilustración 2: Universidad Politécnica de Cataluña, Facultad de Informática de Barcelona</i>	14
<i>Ilustración 3: Universidad de Jaén</i>	15
<i>Ilustración 4: Instituto Tecnológico de Massachusetts, USA</i>	15
<i>Ilustración 5: Universidad de Stanford, USA</i>	16
<i>Ilustración 6: Visita virtual del MIT en "www.youvisit.com"</i>	16
<i>Ilustración 7: Diagrama del "Disp. de Entr. de Tubo de Rayos Catódicos"</i>	17
<i>Ilustración 8: "Nimrod" de John Bennet</i>	17
<i>Ilustración 9: Dr. Dietrich Prinz y su programa de ajedrez</i>	18
<i>Ilustración 10: "OXO" de Alexander S. Douglas</i>	18
<i>Ilustración 11: "Tennis for Two" de William Higginbotham</i>	18
<i>Ilustración 12: "Spacewar!" de Steve Russel</i>	18
<i>Ilustración 13: "Galaxy Game" de B. Pitts y H. Tuck</i>	19
<i>Ilustración 14: "Computer Space" de Nolan Bushnell</i>	19
<i>Ilustración 15: "Magnavox Odyssey" de Ralph Baer</i>	19
<i>Ilustración 16: "Pong" de Al Alcorn</i>	19
<i>Ilustración 17: "Pac-Man" de Namco</i>	20
<i>Ilustración 18: "Battlezone" de Atari</i>	20
<i>Ilustración 19: "Tron" de Midway</i>	20
<i>Ilustración 20: "Nintendo Entertainment System" de Nintendo</i>	20
<i>Ilustración 21: "La abadía del crimen" de Opera Soft</i>	21
<i>Ilustración 22: Nintendo Gameboy</i>	21
<i>Ilustración 23: Sega Megadrive</i>	21
<i>Ilustración 24: "Doom" de id Software</i>	22
<i>Ilustración 25: Sony PlayStation</i>	22
<i>Ilustración 26: Nintendo DS</i>	23
<i>Ilustración 27: Sony PlayStation 2</i>	23
<i>Ilustración 28: "World of Warcraft" de Blizzard</i>	24
<i>Ilustración 29: Logotipo de Autodesk 3ds Max</i>	27
<i>Ilustración 30: Interfaz 3ds Max</i>	27
<i>Ilustración 31: Logotipo de Autodesk Maya</i>	28
<i>Ilustración 32: Interfaz Maya</i>	28
<i>Ilustración 33: Logotipo de Blender</i>	29
<i>Ilustración 34: Interfaz Blender</i>	29
<i>Ilustración 35: Logotipo de CryEngine</i>	30
<i>Ilustración 36: Interfaz CryEngine</i>	30
<i>Ilustración 37: Logotipo de Unreal Engine</i>	31
<i>Ilustración 38: Interfaz de Unreal Engine</i>	31
<i>Ilustración 39: Logotipo de Unity 3D</i>	32
<i>Ilustración 40: Interfaz de Unity 3D</i>	32
<i>Ilustración 41: Teclado y ratón</i>	34
<i>Ilustración 42: Gamepad</i>	35

<i>Ilustración 43: PS Vita</i> .....	35
<i>Ilustración 44: Microsoft Kinect</i> .....	35
<i>Ilustración 45: Oculus Rift</i> .....	35
<i>Ilustración 46: Flujo de trabajo gestionado por el motor de juego</i> .....	37
<i>Ilustración 47: Esquema de la arquitectura de la aplicación</i> .....	42
<i>Ilustración 48: Ed. Betancourt. Plano técnico de la "Planta Sótano"</i> .....	44
<i>Ilustración 49: Ed. Betancourt. Plano técnico de la "Planta Baja"</i> .....	45
<i>Ilustración 50: Ed. Betancourt. Plano técnico de la "Planta Primera"</i> .....	45
<i>Ilustración 51: Ed. Betancourt. Plano técnico de la "Planta Segunda"</i> .....	45
<i>Ilustración 52: Ed. Betancourt. Plano técnico de la "Planta Tercera"</i> .....	46
<i>Ilustración 53: Vista exterior del Ed. Betancourt desde el ed. Sabatini</i> .....	46
<i>Ilustración 54: Vista exterior del Ed. Betancourt desde la cúpula de la biblioteca</i> .....	46
<i>Ilustración 55: Ed. Betancourt. Vista interior de los pasillos</i> .....	47
<i>Ilustración 56: Ed. Betancourt. Vista interior de aulas y laboratorios</i> .....	47
<i>Ilustración 57: Boceto "Menú Principal"</i> .....	48
<i>Ilustración 58: Boceto "Menú Pausa"</i> .....	48
<i>Ilustración 59: Boceto "Menú Cambiar Calidad Gráfica"</i> .....	49
<i>Ilustración 60: Jerarquía de carpetas para Blender</i> .....	50
<i>Ilustración 61: Jerarquía de carpetas para Unity 3D</i> .....	51
<i>Ilustración 62: Resultado final de la visita virtual</i> .....	53
<i>Ilustración 63: Mediciones ed. Betancourt</i> .....	54
<i>Ilustración 64: Creación de paredes, vista planta</i> .....	55
<i>Ilustración 65: Creación de paredes, vista en perspectiva</i> .....	55
<i>Ilustración 66: Planta baja con el suelo sin cortes</i> .....	56
<i>Ilustración 67: Planta primera con el suelo modelado</i> .....	56
<i>Ilustración 68: Columnas de los talleres de la planta baja</i> .....	56
<i>Ilustración 69: Columnas y vigas "amarillas" del sótano</i> .....	57
<i>Ilustración 70: Columnas y vigas cuadradas del sótano</i> .....	57
<i>Ilustración 71: Acabado de columna con y sin suavizado</i> .....	57
<i>Ilustración 72: Detalle de los huecos para las ventanas</i> .....	58
<i>Ilustración 73: Detalle de los huecos para las puertas</i> .....	58
<i>Ilustración 74: Detalle de la creación de las escaleras</i> .....	58
<i>Ilustración 75: Resultado final de las escaleras</i> .....	59
<i>Ilustración 76: Detalle de la escalera principal</i> .....	59
<i>Ilustración 77: Detalle de la sala de maquinaria del ascensor</i> .....	60
<i>Ilustración 78: Detalle de la estructura del techo de los talleres</i> .....	60
<i>Ilustración 79: Detalle de las columnas del soportal</i> .....	60
<i>Ilustración 80: Columnas de la fachada frontal</i> .....	61
<i>Ilustración 81: Detalle sujeciones del enrejado</i> .....	61
<i>Ilustración 82: Rampa de acceso al aparcamiento subterráneo</i> .....	61
<i>Ilustración 83: Distintos tipos de puertas modelados</i> .....	62
<i>Ilustración 84: Tipos de ventanas</i> .....	63
<i>Ilustración 85: Modelado de las cristaleras</i> .....	64
<i>Ilustración 86: Modelado de los ascensores</i> .....	64
<i>Ilustración 87: Barandillas situadas en la pasarela de los talleres</i> .....	64

<i>Ilustración 88: Tipos de luces.....</i>	<i>65</i>
<i>Ilustración 89: Puerta del garaje y barrera de paso .....</i>	<i>65</i>
<i>Ilustración 90: Cubierta metálica del pasamano.....</i>	<i>66</i>
<i>Ilustración 91: Ventana de conserjería.....</i>	<i>66</i>
<i>Ilustración 92: Soporte de carteles .....</i>	<i>66</i>
<i>Ilustración 93: Representación gráfica de la técnica UV Mapping .....</i>	<i>67</i>
<i>Ilustración 94: Mapas de texturas.....</i>	<i>67</i>
<i>Ilustración 95: Ejemplo de UV Mapping aplicado al diseño de una puerta .....</i>	<i>68</i>
<i>Ilustración 96: Creación de la textura "Barrera" .....</i>	<i>69</i>
<i>Ilustración 97: Proceso de creación del suelo de mármol .....</i>	<i>69</i>
<i>Ilustración 98: Generación del mapa de normales con la herramienta de Nvidia .....</i>	<i>70</i>
<i>Ilustración 99: Muestra de diferentes tipos de texturas para los carteles del edificio.....</i>	<i>71</i>
<i>Ilustración 100: Muestras de diferentes tipos de texturas.....</i>	<i>72</i>
<i>Ilustración 101: Muestras de diferentes tipos de texturas.....</i>	<i>73</i>
<i>Ilustración 102: Propiedades de los distintos tipos de "shaders" .....</i>	<i>75</i>
<i>Ilustración 103: Comparativa de la propiedad "Main Color" .....</i>	<i>76</i>
<i>Ilustración 104: Comparativa de renderizado con y sin "occlusion culling" .....</i>	<i>77</i>
<i>Ilustración 105: Comparativa Luz Ambiente .....</i>	<i>79</i>
<i>Ilustración 106: Propiedades iluminación exterior.....</i>	<i>79</i>
<i>Ilustración 107: Iluminación interior. Características del punto de luz .....</i>	<i>80</i>
<i>Ilustración 108: Iluminación interior. Características del foco de luz.....</i>	<i>80</i>
<i>Ilustración 109: Propiedades del controlador de personaje.....</i>	<i>81</i>
<i>Ilustración 110: Interacción con puertas automáticas.....</i>	<i>82</i>
<i>Ilustración 111: Interacción con puertas de apertura manual .....</i>	<i>83</i>
<i>Ilustración 112: Rampa situada sobre las escaleras .....</i>	<i>84</i>
<i>Ilustración 113: Ventana del componente Animation para crear animaciones.....</i>	<i>84</i>
<i>Ilustración 114: Configuración de las animaciones en las propiedades de la puerta .....</i>	<i>85</i>
<i>Ilustración 115: Menú Principal.....</i>	<i>85</i>
<i>Ilustración 116: Menú Pausa con el submenú desplegado .....</i>	<i>86</i>
<i>Ilustración 117: Comparativa entre calidades gráficas.....</i>	<i>87</i>
<i>Ilustración 118: Muestras del resultado final.....</i>	<i>88</i>
<i>Ilustración 119: Muestras del resultado final.....</i>	<i>89</i>
<i>Ilustración 120: Muestras del resultado final.....</i>	<i>90</i>
<i>Ilustración 121: Muestras del resultado final.....</i>	<i>91</i>
<i>Ilustración 122: Planificación estimada de tareas .....</i>	<i>96</i>
<i>Ilustración 123: Planificación estimada de trabajo .....</i>	<i>96</i>
<i>Ilustración 124: Planificación real de tareas .....</i>	<i>97</i>
<i>Ilustración 125: Planificación real de trabajo.....</i>	<i>97</i>
<i>Ilustración 126: Código PEGI .....</i>	<i>102</i>
<i>Ilustración 127: Casos de uso en "Menú Principal" .....</i>	<i>117</i>
<i>Ilustración 128: Casos de uso durante la "Visita Virtual" .....</i>	<i>118</i>
<i>Ilustración 129: Casos de uso en "Menú Pausa" .....</i>	<i>119</i>

# 1. Introducción y objetivos

---

Internet se ha convertido en una parte fundamental de nuestra vida diaria, estando al alcance de cualquier persona en cualquier momento de una manera rápida y sencilla. Con sólo un clic de ratón podemos tener acceso a infinidad de información disponible en las páginas web, con el único inconveniente de que la información obtenida nos muestra un mundo bidimensional, ya que es imposible que a través de fotografías o vídeos se nos muestren las tres dimensiones existentes en el mundo real. Debido a esta carencia, surge la necesidad de investigar cómo se puede integrar la tercera dimensión a la información suministrada. De esta búsqueda nace la realidad virtual.

El objetivo principal de la realidad virtual es conseguir, mediante tecnología informática, el efecto denominado inmersión, que consiste en conseguir introducir a los usuarios en un ambiente virtual y permitir que interactúen con el entorno artificial de la misma manera que si estuviesen realmente en ese sitio. Para conseguir esta sensación, se utilizan diferentes dispositivos que ayudan al usuario a introducirse en esta nueva experiencia, como puede ser el uso de elementos clásicos como el teclado y el ratón, o sistemas de interacción avanzados como gafas de realidad virtual o sistemas de reconocimiento de movimiento.

Paralelamente a la realidad virtual se sitúan las visitas virtuales, donde el objetivo se comparte (conseguir acercarse a las personas lugares en los que no se encuentran en ese instante), pero el sistema para conseguirlo es completamente diferente. Mientras que en la realidad virtual se intenta que el usuario interactúe completamente con el entorno, las visitas virtuales ofrecen una información muy limitada utilizando un recorrido prefijado a través de fotografías o vídeos, pero donde el usuario está atado a las decisiones que tomó el creador de la visita a la hora de elegir los lugares que quiere mostrar, sin posibilidad de conocer el resto del entorno.

De la idea de unificar la libertad absoluta de movimiento e interacción que aporta la realidad virtual y el conocimiento de lugares cercanos y reales ofrecido por las visitas virtuales nace este proyecto, donde se pretende utilizar la tecnología de la realidad virtual de manera que permita conocer al usuario su entorno como hacen las visitas virtuales, sin necesidad de desplazarse de casa. De esta manera, cualquier usuario podrá explorar un entorno que pueda resultar de su interés sin necesidad de viajar físicamente hasta él.

Al querer modelar un emplazamiento real en 3D, de tal manera que se genere una experiencia inmersiva, es necesario crear modelos virtuales y poder interactuar con ellos, por lo que se ha optado por utilizar la tecnología de desarrollo de videojuegos como base para su creación. Como la visita va a utilizar el mismo sistema de movimiento, cámara e interacción que un videojuego, pero con una finalidad informativa en vez de lúdica, se puede considerar que la visita virtual es un videojuego educativo en el que el visitante tendrá libertad total en el entorno generado como ocurre en la realidad virtual.

Pero para la creación de nuestra visita virtual es necesario pensar en los pasos necesarios para crear un videojuego, por lo que hay que tener en cuenta diferentes aspectos antes de realizar el trabajo como son: selección de la tecnología a usar, ya que habrá que saber con qué

herramientas hay que trabajar para conseguir el mejor resultado; bocetos de los edificios y diferentes partes de la escenografía; diseño de los niveles que compondrán la visita, de tal manera que sepamos dónde situar cada elemento; interacción de nuestro visitante con el entorno; cinemáticas y animaciones; y por supuesto, las opciones que queramos que los usuarios tengan disponibles mientras realizan la visita.

Teniendo en cuenta todas estas opciones se podría implementar una visita virtual indistintamente del lugar y permitir que desde cualquier parte del mundo se pudiese acceder a ella. En nuestro caso, se pretende crear una visita virtual de un lugar cercano y conocido, por lo que se ha elegido el edificio Agustín de Betancourt del campus de Leganés de la Universidad Carlos III de Madrid, de tal manera que cualquier persona interesada en acercarse a dicho entorno de estudio pueda conocer de primera mano el ambiente universitario, ya que no se ha encontrado una visita virtual de esas características ni en las facultades más importantes a nivel nacional (como la Politécnica de Madrid o de Barcelona) ni en las internacionales (Universidad de Stanford o Instituto Tecnológico de Massachusetts, referentes a nivel mundial de estudios superiores tecnológicos). A pesar de que algunas de las visitas poseen una gran cantidad de información, apoyando las fotografías panorámicas con textos explicativos, siguen manteniendo un recorrido predefinido por fotografías o videos situados en lugares concretos que no permiten al usuario conocer realmente que se encontraría si llegase a visitar el campus deseado.

Gracias a esto, se pretende acercar el campus de Leganés a cualquier persona interesada en conocerlo, creando además una experiencia innovadora que no se ha encontrado en ninguna facultad del mundo. Así, futuros alumnos que estén interesados en conocer la Universidad como posible lugar de estudios, alumnos actuales que quieran saber dónde están las aulas o despachos antes de desplazarse al edificio, o simplemente personas que quieran acercarse al campus, podrán conocer de primera mano y con total libertad las instalaciones que ofrece la UC3M.

Los objetivos que se pretende desarrollar con la realización de este proyecto son:

- Estudiar las etapas de desarrollo que conlleva realizar una visita virtual a partir de las herramientas existentes en la actualidad para la creación de videojuegos.
- Creación de una visita virtual ubicada en el edificio Betancourt del campus de Leganés de la Universidad Carlos III de Madrid desde cero.
- Incorporación de elementos interactivos que permitan al usuario sentir que se encuentra en el propio edificio.
- La aplicación desarrollada tendrá una finalidad informativa, ya que permitirá a cualquier persona conocer el entorno de estudio universitario.
- También se busca que tenga un propósito educativo, permitiendo tanto a los alumnos como al personal universitario conocer las instalaciones en profundidad.
- Acercar al mayor número posible de personas la sensación de estar visitando el edificio Betancourt utilizando los recursos disponibles a su alcance.

La realización de los objetivos mencionados se detallan en este documento más adelante, de igual manera que los problemas y dificultades surgidos al enfrentarse al trabajo con herramientas en las que no existe conocimiento previo.

Además, se pretende que después de terminar el trabajo, el proyecto tenga diferentes usos tales como servir de publicidad a la Universidad o como soporte de investigación que ayude a mejorar los protocolos de seguridad referentes a evacuaciones en situaciones de emergencia, simulando catástrofes como incendios, inundaciones o derrumbes.

Para terminar este capítulo y ayudar al seguimiento de todo el documento, a continuación se describen los capítulos en los que está dividida la memoria, de tal manera que se posea una visión global del contenido:

**Estado del arte:** Introducción a las visitas virtuales, historia de los videojuegos y estudio de las herramientas para diseño y modelado tridimensional, así como los distintos métodos de interacción usuario-ordenador que existen en la actualidad.

**Definición del problema:** Explicación detallada de los problemas que hay que afrontar y las decisiones tomadas para la realización del proyecto.

**Desarrollo del problema:** Análisis de la aplicación, donde se detallan los requisitos, estructuración, gestión de datos y fuentes consultadas para la realización del proyecto.

**Implementación:** Explicación detalla de la realización de cada elemento modelado y la implementación de todos ellos, junto con la interacción y la lógica del proyecto.

**Pruebas:** Descripción de las pruebas que se han realizado a la aplicación para comprobar su correcto funcionamiento, así como los resultados obtenidos de ellas.

**Planificación y presupuesto:** Detalle de las tareas que componen el proyecto y el tiempo que se necesitará para completarlas. Además se presenta el presupuesto detallado correspondiente al proyecto.

**Marco regulador y entorno socio-económico:** Explicación de las normativas que se deben cumplir para crear un producto de ocio electrónico e impacto socio-económico de la industria del videojuego en la actualidad.

**Conclusiones:** Justificación de la consecución de los objetivos planteados al inicio del proyecto, incluyendo posibles líneas de trabajo futuro.

**Anexos:** Varios documentos con información adicional para completar la información del documento entre los que se encuentran los requisitos del sistema, casos de uso, scripts y manual de usuario.



## 2. Estado del arte

---

En este capítulo se realiza un resumen de la situación actual de las visitas virtuales que se pueden encontrar en las páginas web de las universidades (nacionales e internacionales), además de un completo análisis de la industria de los videojuegos. De la misma manera, también se estudiarán las distintas herramientas disponibles para poder realizar nuestro proyecto, así como los métodos de interacción existentes en la actualidad. De esta forma, el lector podrá valorar de manera óptima el trabajo realizado y así conseguir apreciar los aspectos más importantes del proyecto.

### 2.1. Visitas virtuales

Las visitas virtuales son una forma fácil de conocer un espacio en cualquier dirección (como si el usuario estuviese en el propio lugar), con el objetivo de aumentar la atracción e interés por el lugar que se está visitando. El problema detectado en todos los casos estudiados es que el concepto de "visita virtual" o "tour virtual" se reduce a la interacción mediante vídeos o fotografías panorámicas esféricas de 360º situadas en determinados lugares, a través de las que navegamos usando el ratón, y no se ha estudiado el ir "un paso más allá", como es el caso de este proyecto. Con la técnica actual en la que se basan la mayoría de las visitas virtuales, al usuario sólo se le muestra una pequeña parte de lo que puede llegar a conocer, siendo en la mayoría de los casos una información muy escasa.

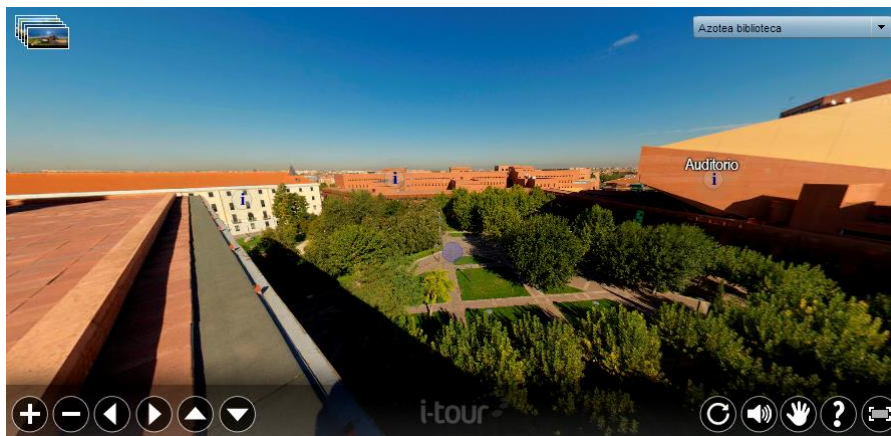
Los contenidos más habituales que se pueden visitar en estos tours interactivos suelen pertenecer a museos, exposiciones, ciudades o universidades, pero usando todos la misma técnica. Por muy completos que quieran mostrarse, siempre están limitados a los contenidos que hayan querido mostrar los creadores, pudiendo ofrecer al usuario una visión muy relativa de lo que pueden llegar a encontrarse.

Las posibilidades de realizar las visitas virtuales se centran en los tours propios de cada web, o en realizar un seguimiento mediante mapas u otras páginas web destinadas a la fotografía. Las opciones más conocidas independientemente de las visitas propias de cada página son las ofrecidas por Google. Mediante Maps o Earth, el usuario puede realizar un recorrido a vista de pájaro (y en determinados sitios, existen representaciones 3D de los edificios), mediante la herramienta StreetView (que consta de fotografías panorámicas 360º cada pocos metros en las calles y de algunos lugares icónicos, como el Parque del Retiro), usando las fotografías geolocalizadas que aporta la gente de manera desinteresada, o con Art Project (centrado en pinacotecas del mundo, recopila imágenes de las obras más importantes y recorridos virtuales por sus galerías). A pesar de todas estas opciones, Google sigue sin mostrar la experiencia interactiva que se busca en este proyecto: la libertad absoluta del usuario.

En el caso de las universidades, y tras navegar por las páginas web de varias instituciones (tanto nacionales como internacionales), se ha comprobado una carencia importante en la libertad del usuario a la hora de conocer los distintos lugares que componen los campus universitarios. En ninguno de los casos estudiados se ha podido apreciar ninguna innovación respecto a las técnicas habituales, ya que la mayoría de los portales que poseían visitas virtuales (que no

todos), basaban la información en mostrar los recintos a través de imágenes panorámicas en 360º, pero sólo de los lugares considerados principales. Por otro lado, algunas universidades recrean la visita a sus instalaciones simplemente mediante un vídeo o fotos estáticas, lo cual minimiza aún más el conocimiento del entorno por parte del visitante. A continuación se muestran las visitas de algunas de las universidades más conocidas (excluyendo aquellas que no dan la posibilidad de realizar el tour):

**Universidad Carlos III de Madrid:** La visita se compone de fotografías panorámicas en 360º de algunos de los lugares más conocidos, dividida por secciones: vistas exteriores (edificios, azoteas, jardines, patios...), vistas interiores (edificios, auditorios, bibliotecas...), y en algunos campus, una tercera sección (laboratorios en Leganés o bibliotecas en Getafe). En todas las fotografías se pueden ver puntos de información en los que se muestran los nombres de los edificios, o acceso a la fotografía correspondiente del lugar visitado. [1]



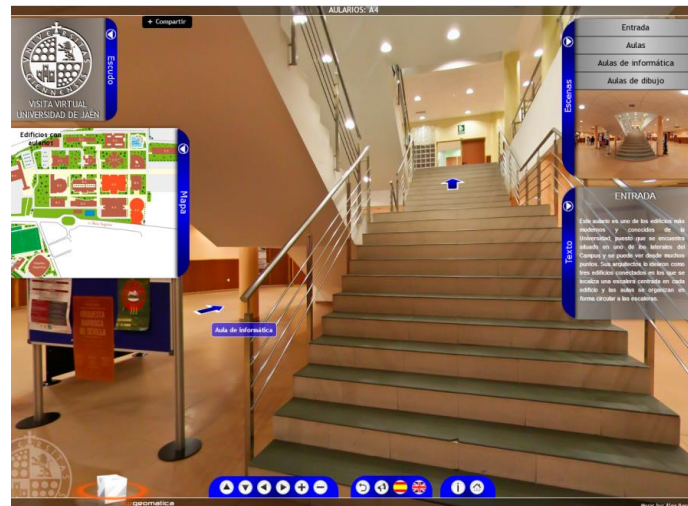
*Ilustración 1: Universidad Carlos III de Madrid, Campus de Leganés*

**Universidad Politécnica de Cataluña (Facultad de Informática):** El portal ofrece varias fotografías panorámicas de varias instalaciones del campus, con la única información del título de la fotografía, la cual indica el lugar en el que se encuentra el usuario. [2]



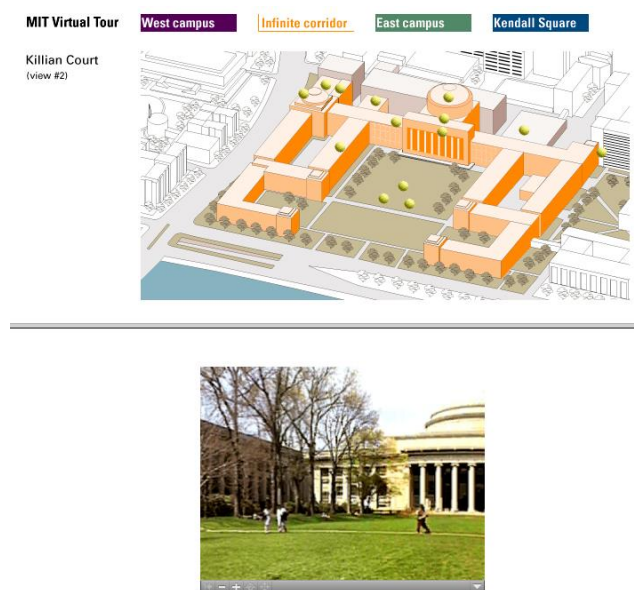
*Ilustración 2: Universidad Politécnica de Cataluña, Facultad de Informática de Barcelona*

**Universidad de Jaén:** La visita virtual más completa que se ha encontrado entre las universidades españolas. Posee un mapa accesible en todo momento en el que se indican tanto los edificios como las localizaciones exteriores que se pueden visitar, indicando mediante flechas las localizaciones próximas que son accesibles, acompañado siempre por un texto explicativo del área en el que se encuentra el usuario. Además, en los interiores de los edificios se ofrece la posibilidad de desplazarse de manera alternativa a las flechas mediante un listado en el que se ven las "escenas" disponibles. Todo ello representado mediante fotografías panorámicas en 360º de muy alta calidad. [3]



*Ilustración 3: Universidad de Jaén*

**Instituto Tecnológico de Massachusetts, USA:** La página web de una de las universidades más importantes a nivel internacional en el ámbito tecnológico nos ofrece un mapa del campus en el que elegir la sección de la Universidad y el lugar de interés que se desea consultar. Tras tener decidido qué visitaremos, se nos ofrece la posibilidad de ver una fotografía panorámica de baja calidad de la zona. Todo ello acompañado por una interfaz gráfica muy pobre y anticuada. [4]



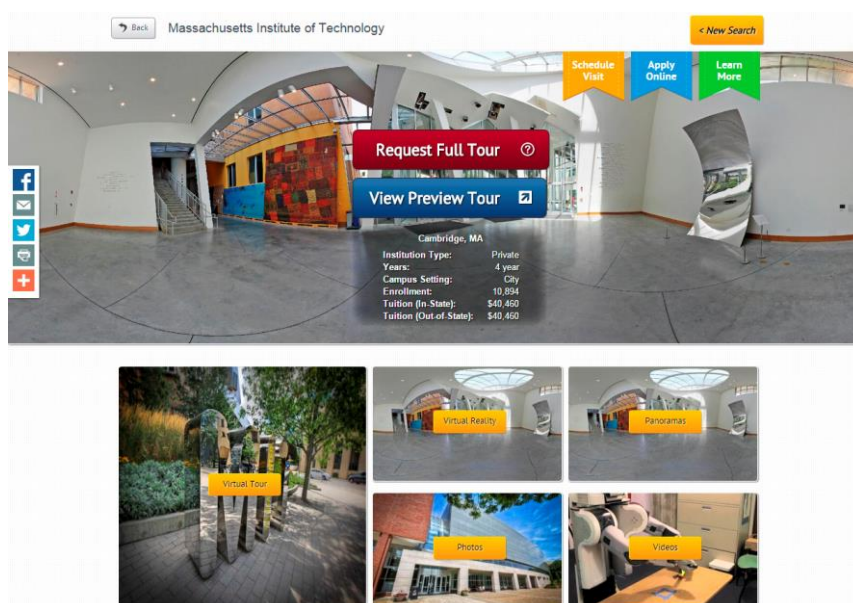
*Ilustración 4: Instituto Tecnológico de Massachusetts, USA*

**Universidad de Stanford, USA:** Otra de las universidades más importantes del mundo, situada en Silicon Valley (California), cuna de las grandes empresas tecnológicas, basa su visita virtual en fotografías acompañadas de un pequeño texto explicativo o vídeos en los que son los propios alumnos los explican la sección que se está consultando. También ofrecen la posibilidad de recorrer el campus mediante un mapa en el que ofrecen la información de todos los edificios y lugares importantes. [5]



*Ilustración 5: Universidad de Stanford, USA*

Llama poderosamente la atención que dos de las universidades más importantes del mundo a nivel tecnológico posean unas visitas virtuales tan escasas de información en sus propios portales. Sin embargo, si se profundiza un poco más, se puede encontrar un tour muchísimo más completo a través de la web "[www.youvisit.com](http://www.youvisit.com)" (especializada en visitas virtuales a los lugares más conocidos del mundo), en la cual existen multitud de fotografías de alta resolución (tanto panorámicas como estáticas), vídeos y audios, que informan al usuario de una manera mucho más completa que en sus propios portales. También llama la atención la inclusión de la posibilidad de usar las gafas de realidad virtual "Oculus Rift" para moverse por las fotografías panorámicas en vez de usar el control habitual basado en el ratón. [6]



*Ilustración 6: Visita virtual del MIT en "www.youvisit.com"*



En este apartado se han explicado los distintos tipos de visitas virtuales existentes, centrado en diversas universidades tanto nacionales como internacionales, (el resto de universidades consultadas poseían tours y visitas muy similares a las mostradas) y se puede concluir que la información mostrada en todas ellas, aparte de seguir el mismo mecanismo, muestran una porción muy pequeña de lo que un posible visitante se encontraría al acercarse al campus real. Por este motivo, para ampliar el concepto de visita virtual, se van a utilizar las técnicas de creación de videojuegos para conseguir llevar una experiencia interactiva a los usuarios.

## 2.2. Industria de los videojuegos

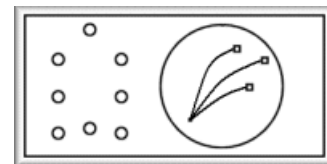
A continuación se realizará un pequeño paseo por la historia de los videojuegos, debido a que la visita virtual va a utilizar las técnicas de creación de un videojuego. Por este motivo, se puede llegar a considerar la visita como un videojuego educativo, ya que el sistema de movimiento e interacción con el entorno utilizados son los mismos que los empleados en la creación de videojuegos.

La industria de los videojuegos es más antigua de lo que se pueda pensar en un principio, remontándose sus orígenes a la década de los años 40, tras la finalización de la Segunda Guerra Mundial. Los primeros "videojuegos" eran experimentos desarrollados por físicos y matemáticos, con carácter de investigación y sin intención de comercializarse ni sacar beneficios de ellos.

Pero antes de hablar de los videojuegos tal y como los conocemos en la actualidad, pondremos al lector en antecedentes sobre los primeros experimentos, juegos electrónicos y juegos gráficos que sentaron las bases de lo que es la industria en la actualidad. [7] [8] [9] [10] [11]

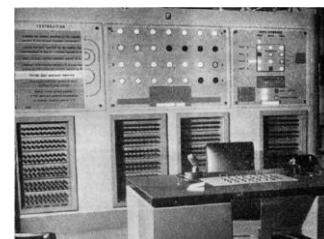
### Los precursores:

**1947:** Thomas T. Goldsmith y Estle Ray Mann crean el "Dispositivo de Entretenimiento de Tubo de Rayos Catódicos", inspirado en los radares militares de la Segunda Guerra Mundial. Este primer "programa" permitía al usuario simular el lanzamiento de misiles contra un objetivo. A pesar de mostrarse en una pantalla CRT (tubo de rayos catódicos) no mostraba ningún gráfico interactivo, y debía ser el usuario quien considerase si se había acertado en el objetivo colocando una lámina semitransparente sobre la pantalla, en la que se veía el itinerario del proyectil. En la *Ilustración 7* se puede ver un diagrama de este dispositivo.



*Ilustración 7: Diagrama del "Disp. de Entr. de Tubo de Rayos Catódicos"*

**1951:** John Bennett presenta "Nimrod", una computadora diseñada para jugar al Nim, un juego matemático de origen chino. La computadora usaba luces para indicar al jugador la situación de la partida, como se observa en la *Ilustración 8*, y aunque es considerado un juego electrónico por carecer de pantalla (y no videojuego), fue la primera computadora diseñada con la única finalidad de "jugar".



*Ilustración 8: "Nimrod" de John Bennet*

**1951:** El Dr. Dietrich Prinz escribió en una computadora "Ferranti Mark I" un programa de ajedrez (*Ilustración 9*) escrito varios años antes (en 1948) por Alan Turing, Claude Shannon y D.G. Champernowne, aunque hasta el año siguiente (1952) no se puso a prueba. El programa simulaba los movimientos de ajedrez (ya que no se mostraban gráficos), los cuales podían tardar entre 15-20 minutos en calcularse. En la segunda partida la máquina ya consiguió su primera victoria. A pesar de ser una simulación sin gráficos y no considerarse un videojuego, sentó las bases de los juegos de ajedrez (que siguen usándose en la actualidad) y se considera el inicio de la Inteligencia Artificial en el mundo de los videojuegos.



*Ilustración 9: Dr. Dietrich Prinz y su programa de ajedrez*

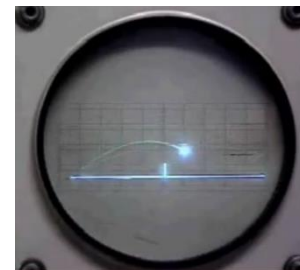
### *Los primeros videojuegos:*

**1952:** Alexander S. Douglas presenta en su tesis una versión computerizada del juego tres en raya llamada "OXO" (También conocido como "Nought and Crosses"). Debido a las múltiples definiciones de "videojuego", hay quienes consideran "OXO" como el primer videojuego de la historia, o simplemente como un juego gráfico por ordenador, *Ilustración 10*.



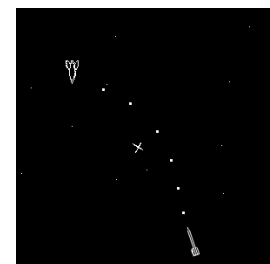
*Ilustración 10: "OXO" de Alexander S. Douglas*

**1958:** William Higginbotham creó un experimento para entretener a los visitantes de su laboratorio que se conocería como "Tennis for Two", el considerado oficialmente primer videojuego de la historia. Mediante un osciloscopio se representaba un partido de tenis para dos jugadores con la vista del campo en perspectiva lateral, incluyendo gravedad para el movimiento de la pelota, *Ilustración 11*.



*Ilustración 11: "Tennis for Two" de William Higginbotham*

**1961:** Steve Russel desarrolla "Spacewar!" en el M.I.T. (Instituto Tecnológico de Massachusetts, USA) en una computadora PDP-1. Es un juego para dos jugadores que consiste en dos naves que combaten en el espacio por destruirse. Este juego es considerado el primer videojuego para ordenador de la historia, *Ilustración 12*.



*Ilustración 12: "Spacewar!" de Steve Russel*

### ***Década de los 70:***

Esta década está marcada por la generalización de los videojuegos, tanto en salones recreativos como domésticamente mediante videoconsolas.

**1971:** La primera versión de "Galaxy Game" de Bill Pitts y Hugh Tuck es comercializada en la Universidad de Stanford (con sólo una máquina construida) para su explotación con monedas en septiembre de este año. El juego era una versión de "Spacewar!". En junio de 1972 se modificó la máquina a su versión "final", la cual estuvo en funcionamiento hasta mayo de 1979, momento en el que fue retirada, *Ilustración 13*.

También en **1971** y en los alrededores de la Universidad de Stanford, pero dos meses después que "Galaxy Game", Nolan Bushnell consiguió comercializar en bares su juego "Computer Space" (*Ilustración 14*), otra versión de "Spacewar!".



*Ilustración 13: "Galaxy Game" de B. Pitts y H. Tuck*



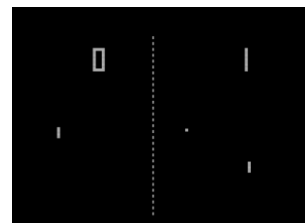
*Ilustración 14: "Computer Space" de Nolan Bushnell*

**1972:** Ralph Baer consigue comercializar la primera consola doméstica llamada "Magnavox Odyssey", que puede verse en la *Ilustración 15*, la cual se conectaba a una televisión y permitía el intercambio entre los 10 juegos que componían su catálogo. De esta manera conseguía terminar el proyecto inicial que comenzó en 1966 con el juego "Fox and Hounds".



*Ilustración 15: "Magnavox Odyssey" de Ralph Baer*

También en **1972** se publicó la máquina recreativa "Pong" de Al Alcorn, bajo el sello de la recién fundada Atari. El videojuego, que se puede considerar la versión comercial de "Tennis for two", consiguió dar el empujón definitivo a la industria del videojuego como fenómeno de masas, *Ilustración 16*.



*Ilustración 16: "Pong" de Al Alcorn*

A partir de ese momento la industria creció exponencialmente con títulos como Breakout (1976), Space Invaders (1978) o Asteroids (1979), y de nuevos sistemas domésticos como Fairchild Channel F (primera consola de cartuchos, 1976), Atari 2600 (1977), Odyssey 2 (1978) o Mattel Intellivision (1979).

### ***Década de los 80:***

Esta época está marcada por la "edad de oro", seguida de la gran crisis (1983-1985) y el nuevo renacer, debido en gran parte a la influencia nipona en el mercado americano y europeo con su nueva visión de los videojuegos.

Desde los últimos años de la década anterior y hasta el año 83, la industria aumentó considerablemente sus ingresos, y su crecimiento e interés por el gran público parecía no tener fin. El mercado se veía invadido por infinidad de máquinas recreativas como el mítico "Pac-Man" (Namco, 1980; *Ilustración 17*), "Battlezone" (Atari, 1980; *Ilustración 18*), "Galaga" (Namco, 1981), "Tron" (Midway, 1982; *Ilustración 19*), o "Zaxxon" (Sega, 1982).



*Ilustración 17: "Pac-Man" de Namco*



*Ilustración 18: "Battlezone" de Atari*



*Ilustración 19: "Tron" de Midway*

Sin embargo, en 1983 un mercado saturado de consolas y miles de videojuegos clónicos unos de otros empezó a resentirse, y la gran crisis estuvo a punto de acabar con la industria debido a la pérdida de confianza de los clientes en una industria en la que habían perdido la fe. A pesar de esta etapa difícil, también se publicó algún juego novedoso y original, como "Tetris" (1984, Alekséi Pázhitnov), que se ha convertido en uno de los juegos más populares que existen.

Tras un par de años muy duros, y en gran medida gracias a Nintendo y su consola "Famicom" (1983; *Ilustración 20*), conocida fuera de Japón como NES (Nintendo Entertainment System) la industria volvió a renacer. La creación de la compañía nipona empezó a abrirse camino en Estados Unidos gracias a juegos innovadores como "Super Mario Bros." (1985), "The Legend of Zelda" (1986) o "Final Fantasy" (1987), los cuales rompían con todos los estándares conocidos hasta la fecha, implementando juegos que no sólo se basaban en obtener una puntuación alta repitiendo la misma pantalla, sino que poseían una historia y finalidad. Sega, por su parte, también entró en el mercado de las consolas con "Master System" (1985), aunque no tuvo mucho éxito.



*Ilustración 20: "Nintendo Entertainment System" de Nintendo*



Mientras tanto, en Europa triunfaban los microordenadores como "Spectrum", "Commodore 64", "Amstrad" o "MSX". Durante estos años se pudo disfrutar de la "edad de oro del software español" (1985-1991), donde los estudios patrios como Indescomp (La Pulga, 1983), Dinamic (Abu Simbel – Profanation, 1985) u Opera Soft (Livingstone Supongo, 1986; La abadía del crimen, 1987; *Ilustración 21*), se encontraban entre las compañías mejor valoradas a nivel europeo.



*Ilustración 21: "La abadía del crimen" de Opera Soft*

### ***Década de los 90:***

Durante los siguientes 10 años se pudo ver como las consolas domésticas se convertían poco a poco en las dominadoras del mercado, junto con la explosión de las consolas portátiles, mientras que las máquinas recreativas, las grandes dominadoras de la década anterior, empezaban a perder su fuerza lenta, pero imparablemente.

La década de los 90 comenzó con la irrupción el año anterior de la primera consola portátil, Gameboy (Nintendo, 1989; *Ilustración 22*), que se convirtió gracias a su precio y amplio catálogo, en la dominadora absoluta año tras año, resistiendo la irrupción de distintas competidoras como Game Gear (Sega, 1990), Lynx (Atari, 1989) o Neo Geo Pocket (SNK, 1998).



*Ilustración 22: Nintendo Gameboy*

Mientras el ámbito portátil comenzaba su guerra particular, las consolas de sobremesa pasaban de la generación de los 8 bits (NES, Master System) a la de 16 bits en 1990, con las consolas Sega Megadrive (Sega Génesis en USA; *Ilustración 23*) y SNES de Nintendo (Super Nintendo Entertainment System). La compañía SNK también intentó entrar en el mercado con Neo Geo, una consola que igualaba las prestaciones (muy superiores) de las máquinas arcade, pero que debido a su alto precio no consiguió la popularidad necesaria.



*Ilustración 23: Sega Megadrive*

Durante los siguientes años, salieron a la venta grandes títulos como Sonic the Hedgehog (Sega, 1991), Lemmings (Psygnosis, 1991), Street Fighter II (Capcom, 1991) o Mortal Kombat (1992). Estos dos últimos juegos, debido a su extrema violencia, provocaron la creación del sistema de calificación de los videojuegos ESRB (Entertainment Software Rating Board) en 1994.

Por su parte, los ordenadores personales iban ganando terreno a los microordenadores, y debido a la mayor potencia que ofrecían, se empezaron a estudiar los entornos tridimensionales, como las

"2D y media" de "Doom" (id Software, 1993; *Ilustración 24*), o los desarrollos completos en 3D como en "4D Boxing" (Distinctive Software, 1991). Esto no significa que las consolas no investigasen también los juegos poligonales, como Virtua Racing (Sega 1992), que marcó un punto de inflexión debido a su gran éxito.



*Ilustración 24: "Doom" de id Software*

A mediados de década, una nueva generación de consolas (esta vez arquitectura de 32 bits) salió a la luz. Tras los intentos fallidos de Sega por aumentar la potencia de Mega Drive con añadidos como Mega CD (1991) y 32X (1994), ésta saca su nueva consola llamada Sega Saturn (1994), y menos de dos semanas después Sony pone a la venta PlayStation (1994; *Ilustración 25*), su primera consola (inicialmente era un añadido a la SNES para soportar CD, pero que debido al fracaso del Sega Mega CD, fue rechazado por Nintendo). Por su parte, Nintendo desarrolla Nintendo 64 (1996), con arquitectura de 64 bits.



*Ilustración 25: Sony PlayStation*

Esta generación queda marcada por el fracaso de Sega Saturn y el triunfo arrollador de Sony PlayStation, convirtiéndose en una de las consolas más vendidas de la historia con un catálogo de grandes éxitos como las sagas Tekken (Namco), Crash Bandicoot (Naughty Dog), Resident Evil (Capcom), Gran Turismo (Polyphony Digital), Metal Gear Solid (Konami) o Final Fantasy (Square). Por su parte, Nintendo 64 quedó relegada al segundo lugar en el mercado, pero manteniendo un gran nivel en su catálogo con juegos como Super Mario 64, Super Smash Bros. y The Legend of Zelda: Ocarina of Time (Nintendo) o GoldenEye 007 y Perfect Dark (Rare).

En PC, tras unos inicios en los que las aventuras gráficas habían marcado la línea a seguir con títulos como "The Secret of Monkey Island" (1990) e "Indiana Jones and the Fate of Atlantis" (1992) de Lucasfilm Games, o "Simon the Sorcerer" (Adventure Soft, 1993), habían evolucionado hacia otro tipo de géneros entre los que se pueden destacar los juegos de estrategia como "Command & Conquer" (Westwood Studios, 1995), "Age of Empires" (Microsoft Games, 1997) o "Starcraft" (Blizzard, 1998), y los FPS (First Person Shooter) como Quake (id Software, 1996), Unreal (Epic, 1998) o Half-Life (Valve, 1998). También hay que destacar en estos últimos años la expansión de las partidas online a través de Internet.

### ***Desde el 2000 hasta la actualidad:***

Con el cambio de milenio la industria del videojuego consiguió afianzarse como una industria multimillonaria, cosa que unos pocos años antes era impensable, consiguiendo e incluso superando las cifras en ventas de otras industrias como el cine o la música. A su vez, las nuevas generaciones de máquinas se implantaron como algo habitual en las casas, desbancando por completo a las máquinas recreativas como los principales centros de entretenimiento y estreno de las nuevas producciones.

Las máquinas arcade estaban en crisis desde mediados de la década anterior, quedando relegadas en la actualidad a muy pocos salones recreativos, y en casi su totalidad, sustentados

por las propias marcas que producen los juegos. La hegemonía siempre se había basado en la superioridad de su tecnología y potencia gráfica, pero con la entrada de las nuevas consolas y ordenadores personales, que igualaban e incluso superaban sus características, empezaron su lento pero imparable camino hacia su casi total desaparición.

En el ámbito portátil seguía predominando con autoridad Gameboy, y no fue hasta 2004 cuando salió la nueva generación de consolas portátiles. Nintendo, tras varias iteraciones de la consola original en los años anteriores (Gameboy Pocket en 1996, Gameboy color en 1998 y Gameboy Advance en 2001), revolucionó el diseño de su baluarte portátil con la nueva Nintendo DS (*Ilustración 26*), una consola con doble pantalla en vertical (una táctil y otra clásica). Sony, por su parte, entró en el mercado a los pocos días con Sony PSP (PlayStation Portable), una consola más potente y con capacidad de reproducción de audio y video, así como conexión a Internet.

Tras la derrota de Sony por Nintendo en las consolas portátiles los años anteriores, ambas compañías volvieron a desarrollar nuevo hardware en 2011. Nintendo apostó por 3DS, una revisión de su anterior consola, con la posibilidad de ver videojuegos en 3D sin necesidad de gafas especiales, mientras que Sony hizo lo mismo con PlayStation Vita, apostando por la potencia gráfica y distintas características como la conexión a sus "hermanas mayores", PlayStation 3 y 4. A pesar de estos intentos de recuperar el mercado portátil por parte de Sony, la historia se repite y Nintendo sigue copando las ventas de dispositivos portátiles.



*Ilustración 26: Nintendo DS*

En entorno doméstico, la sexta generación (con arquitectura de 128 bits) estaba a punto de darse a conocer. A pesar de que en 1998 Sega había lanzado Dreamcast, fuera de Japón no había tenido mucha difusión, y no fue hasta el año 2000, cuando Sony lanzó la esperada PlayStation 2 (*Ilustración 27*), cuando el público empezó a dar el salto hacia las nuevas consolas. Microsoft entró en la industria, igual que Sony en la generación anterior, con su consola Xbox, de características similares al hardware de Sony. Nintendo, por su parte, también sacó nueva consola el mismo año que Microsoft, denominada GameCube. Tanto Xbox como GameCube, a pesar de ser buenas consolas, no tuvieron el éxito arrollador cosechado por PlayStation 2.



*Ilustración 27: Sony PlayStation 2*

Pero no acaban ahí los avances, ya que en el año 2005, Microsoft se adelanta a sus competidores y pone a la venta Xbox 360, la primera consola de la séptima generación. Al año siguiente, Sony con PlayStation 3 y Nintendo con su novedosa Nintendo Wii (basando sus controles en un mando con acelerómetros y sensores que reproduce los movimientos del jugador). Durante estos años, Sony pierde su hegemonía absoluta, y la industria se iguala en ventas.

Por último, la octava generación de consolas es iniciada por Nintendo con Wii U en 2012, mientras que Microsoft con Xbox One y Sony con PlayStation 4 se suman a esta nueva iteración de consolas un año más tarde, en 2013.

Por su parte, el mercado de los ordenadores viene marcado por la expansión de los juegos online, tanto de acción en primera persona con sagas como Call of Duty (Activision) o Battlefield (EA), como de estrategia en tiempo real con las series de juegos Warcraft (Blizzard) o Rise of Nations (Microsoft Games). Paralelamente se creó un nuevo género, los MMORPG (Massively multiplayer online role-playing game). Estos juegos consisten en ponerse en la piel de un personaje en un mundo virtual y hacerle evolucionar y progresar a lo largo de su aventura interactuando con otros jugadores humanos a través de Internet. El juego que consiguió popularizar este género fue Ultima Online (EA, 1997), al que siguió Lineage (NCSOFT, 1998), y el archiconocido World of Warcraft (Blizzard, 2004; *Ilustración 28*), entre muchos otros.



*Ilustración 28: "World of Warcraft" de Blizzard*

Tras realizar un estudio de las visitas virtuales y la historia de los videojuegos para poder comprender el entorno en el que vamos a trabajar, es necesario analizar las herramientas disponibles de modelado tridimensional y de implementación mediante el motor de juego, por lo que se profundizará en las características principales del software existente para poder comprobar qué programas se ajustan mejor a las necesidades del proyecto.

### 2.3. Diseño 3D

El diseño 3D es el campo de la informática dedicado a la creación de modelos tridimensionales generados por ordenador. Debido al incremento de la demanda de esta tecnología en otros ámbitos fuera de la computación, es uno de los campos con mayor evolución y progresión, debido a que con el aumento de potencia en los equipos informáticos se consiguen realizar modelos cada vez más realistas.

Aunque en este caso en particular se utiliza su aplicación para la implementación en videojuegos, el diseño 3D se usa en infinidad de ámbitos distintos al que se va a tratar. El más conocido es la aplicación al cine, pero en medicina se usa para realizar modelos detallados de los órganos, en el sector científico se utiliza para la representación de componentes químicos, en arquitectura para mostrar los diseños de obras y remodelaciones en edificios y parques, y en ingeniería para dar a conocer vehículos y estructuras, o diseñar maquinaria y sus componentes.

Volviendo al terreno de los videojuegos, el diseño se compone de la fase de modelado (centrada en la creación de objetos y modelos), y la fase de implementación mediante el motor gráfico de videojuego (centrada en la composición final de la escena a mostrar y la implementación de la interacción con el usuario).

El modelado es la fase del desarrollo en la que se crean los objetos que deben ser usados posteriormente. Se debe prestar extrema atención y cuidado a los modelos generados, ya que esta etapa es la base fundamental de cualquier videojuego, invirtiéndose la mayoría de tiempo y recursos.

La elección del motor de juego es una de las decisiones más importantes a la hora de llevar a cabo un videojuego, ya que en función del motor que elijamos, el resultado final puede variar enormemente. Existen motores enfocados a realizar juegos para ordenador o videoconsolas, a juegos de mundo abierto, a plataformas, o con potencia suficiente para mostrar gráficos realistas... y en función de la mecánica y apariencia que queramos obtener, debemos saber que elegir. Pero... ¿qué es exactamente un motor de juego?

Un motor de juego es un término que hace referencia a un conjunto de rutinas de programación que otorgan la capacidad de diseñar y representar un videojuego. La funcionalidad básica es dotar al juego de la posibilidad de renderizar gráficos, físicas, detección de colisiones... pero con la evolución constante del hardware y del software, los motores han conseguido completar sus funciones nativas con infinidad de opciones.

De esta manera, permiten fácilmente la creación de juegos muy completos, ya que a las funcionalidades ya nombradas (renderizado, físicas y colisiones), han ido añadiendo opciones básicas para desarrollar los juegos, como es la iluminación dinámica, editores de sonido, animaciones, scripting, inteligencia artificial, sistemas de partículas, gestión de redes y streaming, administración de memoria, o uso de librerías de tratamiento de gráficos (como OpenGL y DirectX), y todo ello con interfaces sencillas e intuitivas que agilicen el trabajo. Con la incorporación de todas estas características, se consiguen aunar en un solo sistema todos los componentes necesarios para crear el proyecto.

Normalmente, los motores de juegos constan de los siguientes componentes:

- **Motor gráfico:** Muestra en pantalla el aspecto visual del juego. Se encarga del renderizado (2D/3D) mapeando las texturas, aplicando efectos como culling o antialiasing, generando las mallas de los objetos 3D o gestionando el número de polígonos en escena.
- **Motor de físicas:** Calcula, gestiona y aplica las leyes físicas del mundo real al mundo virtual del videojuego de la manera más fiel posible. Se puede apreciar en la gestión de la gravedad, masas, fuerza, velocidad, fluidos, etc.
- **Detector de colisiones:** Sistema encargado de calcular constantemente la posición de cada objeto que hay en escena para comprobar si están en contacto. Es usado para evitar que los objetos se atraviesen, activar acciones al pasar por un punto determinado, etc.

- **Motor de inteligencia artificial:** Es uno de los módulos más importantes y complicados del motor. Se encarga de dotar al sistema de "realismo", programando las reacciones y comportamiento de los NPC (Non Player Characters - Personajes No Jugables) ante los estímulos que le afectan.
- **Motor de sonido:** Encargado de gestionar, controlar y reproducir todos los sonidos de la escena (música, efectos sonoros, diálogos).
- **Gestor de redes:** Componente encargado de la gestión y sincronización en línea. Se utiliza principalmente para gestionar las partidas online.

Ante tal cantidad de información distinta, pero a la vez tan importante, surge la duda de qué motor usar para la creación de nuestro videojuego.

Dada la magnitud del proyecto, se necesita tanto un programa de modelado como un motor de videojuegos potentes, que puedan gestionar la gran cantidad de información que se necesitará crear y gestionar, por lo que se estudiarán las opciones profesionales que se encuentran en el mercado, y que a la vez se consideren más apropiadas para el tipo de proyecto realizado. Las herramientas estudiadas, aparte de por la potencia ofrecida, serán seleccionadas en base a los recursos disponibles, comunidad que los apoya, ejemplos, resultados que ofrecen y curva de aprendizaje.

A continuación realizaremos un breve análisis de los programas que más se ajustan a las necesidades del trabajo que se va a realizar, contando las características más importantes y su modelo de negocio, terminando con una tabla comparativa de cada uno de ellos.



### 3ds Max

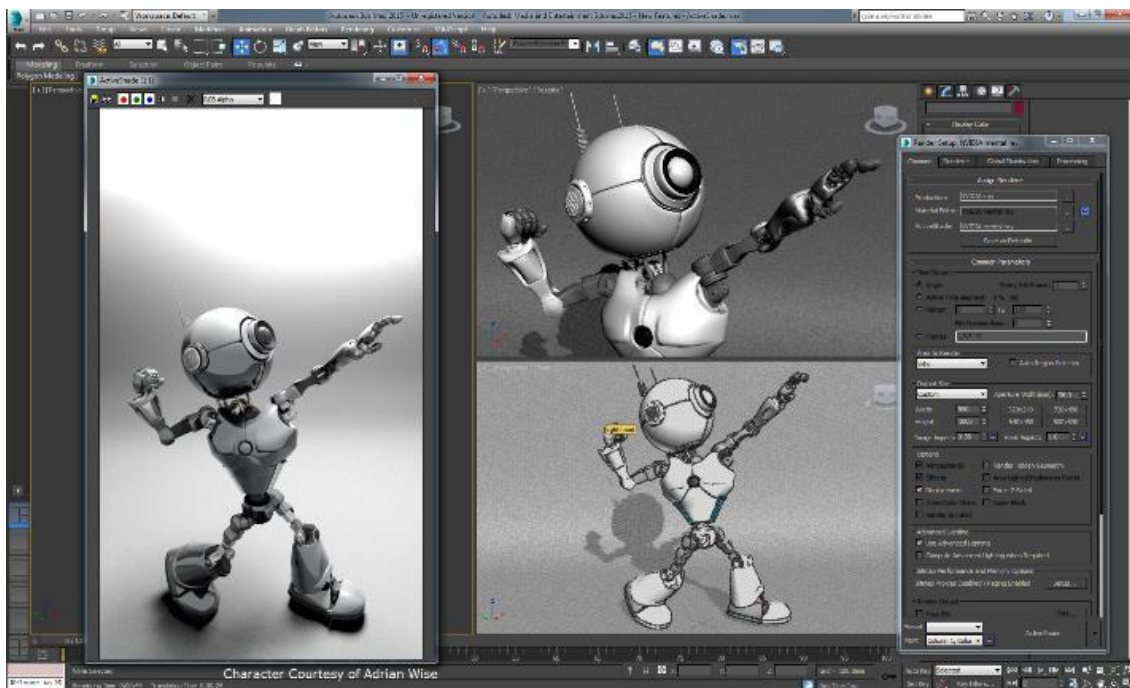
Autodesk 3ds Max es un software de modelado, animación y renderizado 3D enfocado a profesionales del diseño y el entretenimiento. Su primera versión se comercializó en 1990 para el sistema operativo DOS, y ha estado en constante evolución, pudiendo conseguirse actualmente la versión 2016, disponible para Windows.



*Ilustración 29: Logotipo de Autodesk 3ds Max*

La arquitectura interna de 3ds Max se basa en el uso de plug-ins, característica que unida a la potencia que posee y su dilatada experiencia, le ha convertido en el programa referente para los desarrollos de videojuegos, la publicidad, la arquitectura y el cine. Al ser un programa enfocado a profesionales, la curva de aprendizaje no es sencilla, pero para paliar este pequeño escollo, posee tutoriales y foros oficiales, consiguiendo resultados realmente increíbles.

3ds Max ofrece una versión de prueba de 30 días, además de una versión gratuita para estudiantes para uso no comercial. En el ámbito comercial ofrece suscripciones desde 195€/mes o 1.560€/año, o licencia autónoma desde 3.900€. [12]



*Ilustración 30: Interfaz 3ds Max*

## Maya

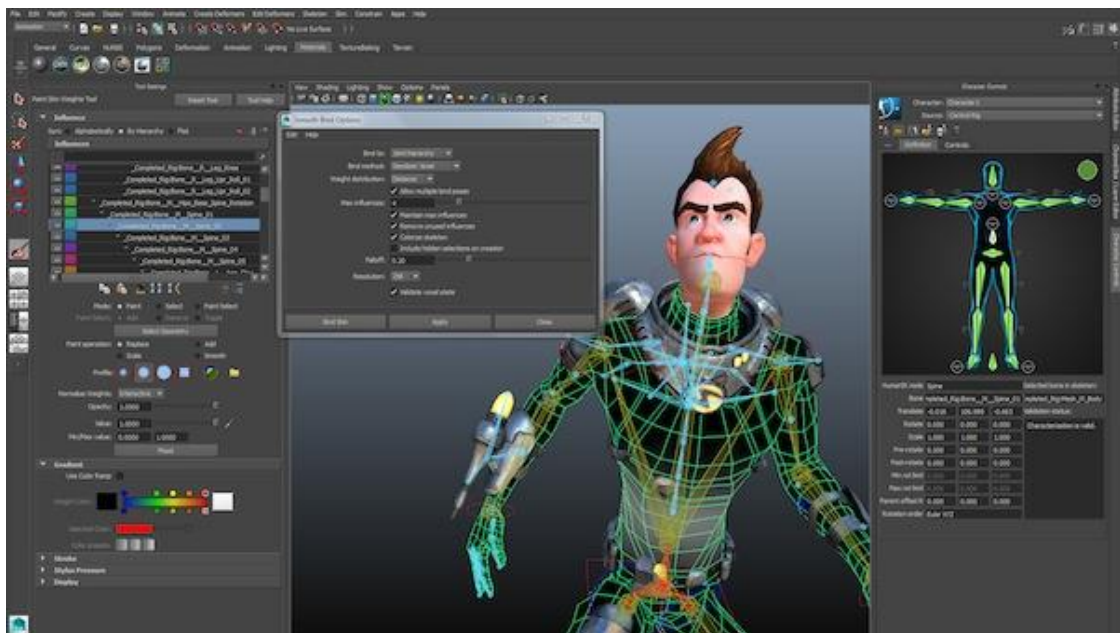
Autodesk Maya es un programa de modelado, animación, simulación y renderizado 3D, enfocado a ofrecer funciones creativas en una plataforma de producción ampliable, proporcionando un aumento de la productividad en modelado, texturización e iluminación. Su primera versión se lanzó en 1998, y en la actualidad se comercializa la versión 2016 en entornos Windows, Mac OS, Red Hat y CentOS.



*Ilustración 31: Logotipo de Autodesk Maya*

Se caracteriza por la potencia y posibilidades de expansión y personalización de su interfaz, además de poseer herramientas nativas especializadas en simulación de fluidos, tejidos y cabello. Estas opciones han conseguido que se haya convertido en el programa predilecto en la industria cinematográfica para realizar efectos especiales, y también para la realización de películas de animación (en gran parte por su compatibilidad con el programa de renderizado fotorrealista RenderMan, perteneciente a Pixar).

Las licencias de uso de Maya son las mismas que para 3ds Max, tanto en las versiones de prueba, para estudiantes o comerciales. [13]



*Ilustración 32: Interfaz Maya*



## Blender

Blender es un programa enfocado principalmente en modelado 3D con licencia Open Source bajo el desarrollo de la Fundación Blender. Publicado por primera vez en 1995, este programa ofrece en la actualidad la última versión publicada (2.74) de manera gratuita en su web, aunque está en continua evolución, sacando versiones nuevas cada 3-4 meses. Soporta los tres sistemas operativos principales (Windows, Mac OS y Linux), lo cual unido a su filosofía de software libre y una comunidad inmensa de seguidores, le convierte en uno de los programas más usados.



Ilustración 33: Logotipo de Blender

Las principales características de esta herramienta son su amplio espectro de funcionalidad tanto para modelado, texturización, iluminación, animación, sincronización de vídeo y edición de audio, renderizado potente, simulaciones dinámicas de fluidos y partículas, lenguaje de programación Python que permite controlar distintas herramientas simultáneamente... y todo ello en un programa que ocupa muy poco espacio en disco duro. [14]

Como característica adicional, posee un motor de juegos basado en OpenGL. Ofrece un entorno visual cómodo a la hora de diseñar aplicaciones interactivas sin necesidad de conocer ningún lenguaje de programación (aunque también ofrece la posibilidad de programar scripts mediante Python). A pesar de éstas características y su gran versatilidad (como realizar todo el trabajo con el mismo programa), está muy por debajo en potencia, funcionalidades y características de lo que ofrecen otros motores de juego, por lo que no se considerará Blender como herramienta de estudio en este ámbito, quedando exclusivamente para modelado.

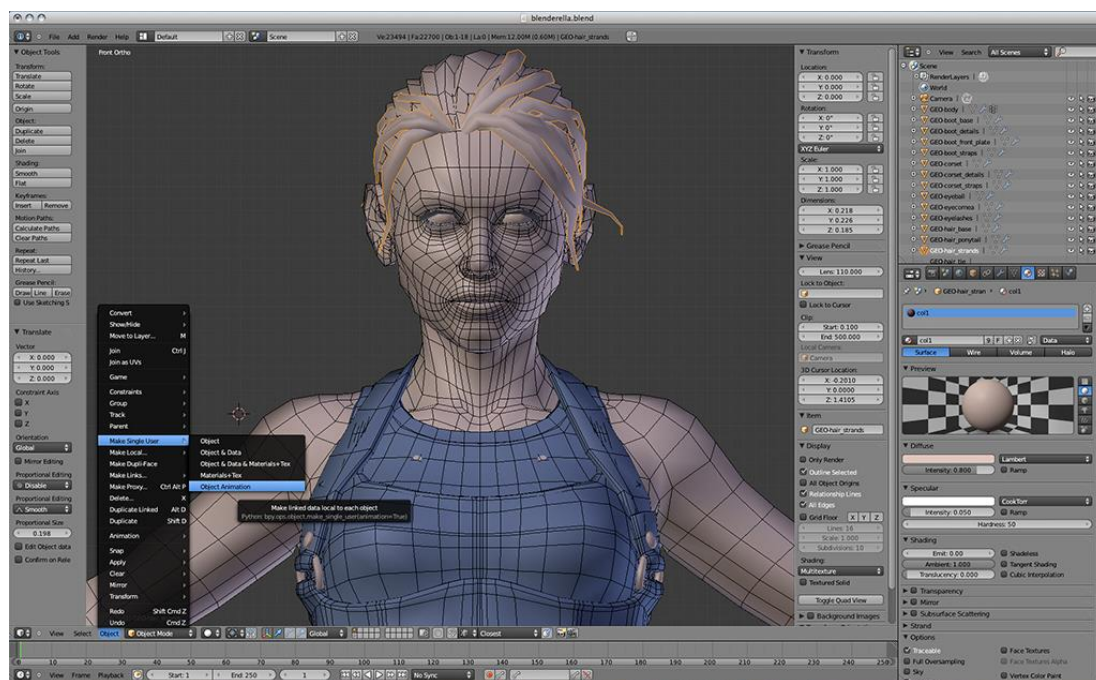


Ilustración 34: Interfaz Blender

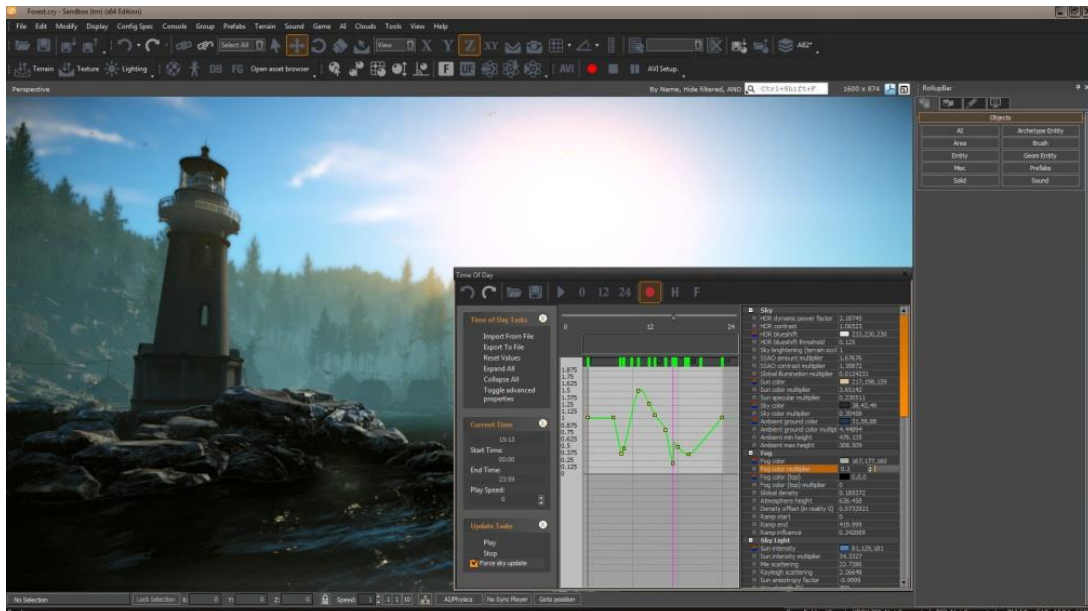
## CryEngine

Creado por la empresa de software Crytek, CryEngine se desarrolló inicialmente como motor de demostración para la empresa Nvidia, pero dado su gran potencia se adapta al desarrollo de videojuegos. Los grandes exponentes de la potencia de este motor son los juegos de la saga Crysis, que asombraron al mundo entero por su espectacularidad. En el año 2006, los derechos de CryEngine son adquiridos por la empresa francesa Ubisoft.



El primer videojuego comercializado utilizando esta tecnología fue Far Cry en el año 2004. Actualmente está disponible la versión CryEngine3 de manera gratuita para proyectos sin ánimo de lucro. Si por el contrario se desea realizar un proyecto comercial, poseen un modelo de pago por suscripción de 9,90€/mes. Para obtener acceso a la versión completa es necesario consultar directamente con la empresa [15]. Las principales características de CryEngine son:

- Actualización en tiempo real en el editor sandbox.
- Soporte multinúcleo
- Iluminación dinámica en tiempo real
- Editor de animación facial
- Motor de físicas completo (agua, vegetación, efectos naturales del mundo, etc.)
- Renderizado de texturas de alta velocidad
- Compatible con PC, consolas y dispositivos móviles.



*Ilustración 36: Interfaz CryEngine*

## Unreal Engine

Creado por la empresa Epic Games, Unreal Engine se desarrolló inicialmente para el videojuego Unreal en 1998, un shooter en primera persona. Dada su gran versatilidad, se ha convertido en el motor gráfico más popular entre las grandes compañías de videojuegos como Activision, Konami EA o Sony entre otros.

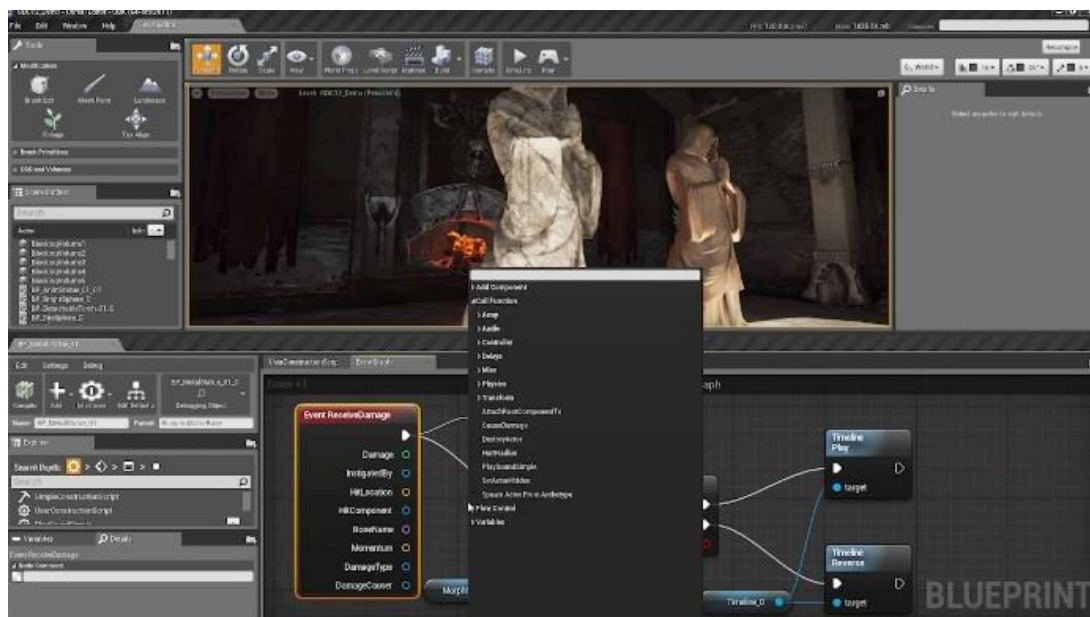
Desde su primera versión, y hasta el actual Unreal Engine 4, este motor ha dado lugar a grandes éxitos de ventas como la saga Gears of War, Batman: Arkham Asylum. Mass Effect o Bioshock.

Su modelo de negocio se basa en una distribución y uso gratuito para estudiantes. Si el uso es comercial, se deberá pagar un 5% de las ganancias generadas siempre y cuando los beneficios por trimestre superen los 3.000\$/producto [16]. Las principales características de Unreal Engine son:

- Sistema de renderizado multihilo, con sistema de iluminación, sombras dinámicas y oclusión.
- Editor completo de vegetación.
- Sistema de creación de scripts de manera gráfica a modo de elementos en un plano que se interrelacionan entre sí.
- Editor de cinemáticas.
- Unreal Script, lenguaje propio similar a Java o C#.
- Compatible con PC, consolas y dispositivos móviles.



*Ilustración 37: Logotipo de Unreal Engine*



*Ilustración 38: Interfaz de Unreal Engine*

## Unity 3D

Desarrollado por Unity Technologies, el motor gráfico Unity 3D es un motor de videojuegos de gran calado entre la comunidad debido a su sencillez de uso y su gran potencia, así como por la opción de ser usado tanto en Windows como en Mac OS. Permite crear juegos multiplataforma tanto en PC, en consolas, dispositivos móviles, y mediante un plug-in, en navegadores.



Ilustración 39: Logotipo de Unity 3D

Los juegos más punteros son Nihilumbra (de la desarrolladora española BeautiFun Games), Kerbal Space Program, y el exitoso Hearthstone: Heroes of Warcraft, de Blizzard.

Su modelo de negocio se basaba (hasta la versión 5, publicada en marzo de 2015) en una licencia gratuita (con algunas características bloqueadas), y licencia comercial (Unity Pro) con versión de prueba de 30 días o completa bajo pago de 1.140€. A partir de la versión 5, el programa se ofrece completo y gratuito para proyectos no comerciales, con un precio de 1.140€ para licencia completa y de 57€/mes para las suscripciones [17]. Las principales características de Unity son:

- Programa con mayor comunidad en internet.
- Interfaz intuitiva y amigable.
- Actualización automática en tiempo real de los objetos usados en el proyecto sin necesidad de re-importar.
- Shading basado en la física.
- Perfiles de memoria avanzados.
- Publicación en Windows, Mac OS, Linux, navegadores de internet (Windows y Mac), consolas y dispositivos móviles.
- Soporta los lenguajes de programación C#, JavaScript y Boo (inspirado en Python).

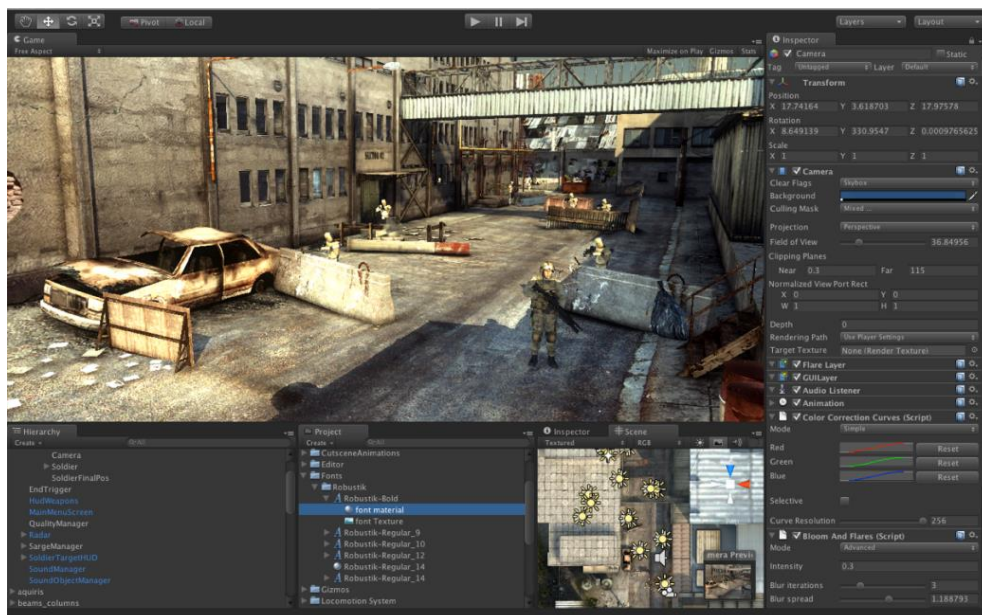


Ilustración 40: Interfaz de Unity 3D



En las siguientes tablas se mostrarán unan pequeñas comparativas entre los software de modelado (Tabla 1) y los motores gráficos (Tabla 2) analizados anteriormente:

Programa	3ds Max	Maya	Blender
<b>Desarrollador</b>	Autodesk	Autodesk	Fundación Blender
<b>Web</b>	<a href="http://autodesk.com/3dsmax">autodesk.com/3dsmax</a>	<a href="http://autodesk.com/maya">autodesk.com/maya</a>	<a href="http://blender.org">blender.org</a>
<b>Sistema Operativo</b>	Windows 7 y 8	Windows (7 y 8.1), Mac OS X, Red Hat y CentOS	Windows, Mac OS, GNU/Linux
<b>Licencia</b>	Propietario	Propietario	Gratuita
<b>Tamaño en disco</b>	6 Gb	4 Gb	230 Mb
<b>Memoria RAM</b>	4 Gb (8 Gb recomendado)	4 Gb (8 Gb recomendado)	2 Gb
<b>Precio</b>	Prueba de 30 días Estudiantes: Gratis Suscripción (desde): ▪ Mensual: 195€ ▪ Anual: 1560€ Licencia autónoma: ▪ desde 3900€	Prueba de 30 días Estudiantes: Gratis Suscripción (desde): ▪ Mensual: 195€ ▪ Anual: 1560€ Licencia autónoma: • desde 3900€	Gratis

Tabla 1: Comparativa programas de modelado

Motor de juego	CryEngine	Unreal Engine	Unity 3D
<b>Desarrollador</b>	Crytek	Epic Games	Unity Technologies
<b>Web</b>	<a href="http://cryengine.com">cryengine.com</a>	<a href="http://unrealengine.com">unrealengine.com</a>	<a href="http://Unity3D.com">Unity3D.com</a>
<b>Sistema Operativo</b>	Windows (Vista, 7 y 8)	Windows (7 y 8) Mac OS X 10.9	Windows (XP, 7, 8) Mac OS X 10.8
<b>Licencia</b>	Propietario	Propietario	Propietario
<b>Lenguaje Scripting</b>	LUA	Unreal Script	C#, JavaScript, Boo
<b>Plataformas de desarrollo</b>	<ul style="list-style-type: none"> <li>▪ PC (Windows)</li> <li>▪ Xbox (360/One)</li> <li>▪ PlayStation (3, 4)</li> <li>▪ WiiU</li> <li>▪ Móviles (iOS, Android)</li> </ul>	<ul style="list-style-type: none"> <li>▪ PC (Windows, Mac, Linux)</li> <li>▪ Xbox (360/One)</li> <li>▪ PlayStation (3, 4)</li> <li>▪ Wii</li> <li>▪ Móviles (iOS, Android)</li> <li>▪ Soporta Realidad Virtual (VR)</li> </ul>	<ul style="list-style-type: none"> <li>▪ PC (Windows, Mac, Linux, navegadores)</li> <li>▪ Xbox (360/One)</li> <li>▪ PlayStation (3, 4, Vita)</li> <li>▪ WiiU</li> <li>▪ Móviles (iOS, Android, Blackberry, Windows Phone)</li> <li>▪ Soporta VR</li> </ul>
<b>Tamaño en disco</b>	8 Gb	-	4.5 Gb
<b>Memoria RAM</b>	8 Gb	8 Gb	-
<b>Precio</b>	Suscripción: <ul style="list-style-type: none"> <li>▪ Mensual 9,90€/mes</li> </ul> Completa: <ul style="list-style-type: none"> <li>▪ Consulta con Crytek</li> </ul>	Gratuito <ul style="list-style-type: none"> <li>▪ Uso académico</li> </ul> Versión comercial: <ul style="list-style-type: none"> <li>▪ 5% beneficios si las ganancias superan 3.000\$/producto por trimestre</li> </ul>	Gratuito (no comercial) Versión Pro: <ul style="list-style-type: none"> <li>▪ Prueba 30 días</li> <li>▪ Completa: 1.140€</li> <li>▪ Suscripción 57€/mes</li> </ul>

Tabla 2: Comparativa motores de videojuegos

### Otros programas

A parte de la sección de modelado y de motor gráfico, muchas veces también es necesario usar otros tipos de programas que pueden pasar más desapercibidos en primera instancia, pero que son igualmente necesarios a la hora de llevar a buen puerto un proyecto de este tipo, como programas de edición de vídeo, audio o imagen.

Esto es debido a que no todos los recursos usados se pueden usar tal y como se consiguen. Si se quiere realizar un video-demostración, es necesario un programa de edición de video en el que se pueda realizar el montaje. Puede ser necesario que haya que separar los efectos sonoros porque vengan distintos efectos en el mismo archivo, o que haya que modificar su duración. De la misma manera, y como ha sido el caso de este proyecto, muchas texturas han sido necesarias modificarlas, o incluso crearlas desde cero.

En este proyecto, la duda a la hora de elegir el programa para la edición de imagen estaba entre el uso de Gimp o Adobe Photoshop. Al ser una herramienta de "apoyo", no vamos a realizar un análisis completo como en los casos anteriores, pero si es importante conocer los motivos por los que se ha optado por trabajar con un software en concreto.

Ambos programas son muy similares (se puede considerar que Gimp es la alternativa Open Source y gratuita de Photoshop, el cual es un programa comercial), además de ser muy potentes y versátiles, con características parecidas y una interfaz de trabajo semejante, pero la elección final ha recaído sobre el programa Photoshop. Esta decisión viene marcada por la experiencia previa en el uso de este programa.

## 2.4. Interacción

En la actualidad existen múltiples formas de interactuar con los videojuegos, de tal manera que el usuario sea capaz de ver plasmado en la pantalla las acciones que quiere que sean realizadas en el entorno virtual. A continuación explicaremos las formas de interacción más extendidas entre el público para controlar los videojuegos, pasando desde el tradicional teclado hasta los novedosos detectores de movimiento. [18]

**Teclado:** Adaptado de las máquinas de escribir al ordenador, el teclado es el método de interacción más común y el preferido por la mayoría de los jugadores de ordenador, debido a su versatilidad para adaptarse a todo tipo de juegos.

**Ratón:** Debido a la creación de las interfaces en los ordenadores, el mundo de la computación se vio en la necesidad de crear un dispositivo apuntador, que facilitase y agilizase la interacción. Fruto de esta necesidad, en 1963 se inventó el ratón, permitiendo a los usuarios definir puntos sobre el espacio mostrado en la pantalla. Se ha convertido en el periférico inseparable del teclado, convirtiéndose en el principal conjunto de entrada de datos. En la *Ilustración 41* se observa en tándem teclado-ratón.



*Ilustración 41: Teclado y ratón*

**Mando:** También conocido como gamepad (*Ilustración 42*), data de la época del videojuego Spacewar! y consiste en un periférico que consta de palancas y múltiples botones, adaptados a la forma de la mano del usuario para facilitar su uso y minimizar el cansancio. Se pueden usar tanto en consolas como en ordenadores, y de la misma manera que la combinación teclado/ratón, su uso suele estar adaptado a todos los géneros de videojuegos.



*Ilustración 42: Gamepad*

**Pantalla táctil:** El usuario, haciendo contacto sobre una pantalla directamente, permite la entrada de datos y las órdenes al dispositivo, a la vez que la pantalla le muestra el resultado introducido. Este contacto se puede realizar mediante "lápices ópticos" o con los dedos directamente. Esta tecnología se puede encontrar en consolas (Nintendo DS, PSVita; *Ilustración 43*), en teléfonos inteligentes o tabletas, incluso en monitores de ordenador.



*Ilustración 43: PS Vita*

**Detección de movimiento:** En los últimos años se ha perfeccionado la técnica de la detección de movimiento para su adaptación a los videojuegos. Utilizando una cámara enfocada al usuario, es capaz de detectar los movimientos que realiza y reflejarlos en la pantalla. Aunque este tipo de interacción existe hace muchos años en los ordenadores usando las webcam, gracias a la inversión realizada por Sony con EyeToy y a Microsoft con Kinect (*Ilustración 44*), el uso de esta tecnología enfocada a videojuegos consiguió extenderse entre el público.



*Ilustración 44: Microsoft Kinect*

**Realidad Virtual:** Utilizando unas gafas o un casco, se reproducen imágenes sobre una pantalla muy cercana a los ojos, ocupando el campo de visión del usuario de modo que no tenga percepción del entorno que lo rodea. Además, el dispositivo cuenta con sensores de movimiento para mover la escena según los movimientos del usuario consiguiendo una inmersión total en el entorno virtual. El punto negativo a esta tecnología, es que suele ser necesario acompañar el movimiento del personaje con otro tipo de periférico, como teclado, mandos, o detectores de movimiento. En la *Ilustración 45* se pueden observar las gafas de realidad virtual Oculus Rift.



*Ilustración 45: Oculus Rift*

## 2.5. Conclusiones

Después de haber realizado un estudio de las distintas opciones que existen para la realización de un videojuego, vamos a elegir las herramientas más apropiadas para realizar el proyecto actual, siguiendo el mismo orden utilizado anteriormente para los análisis.

### *Modelado*

Los tres programas estudiados poseen características similares en el ámbito de uso que vamos a necesitar, como es el diseño de objetos con mapeado y texturizado, y con múltiples opciones de exportación e importación, compatibles con los motores gráficos estudiados.

A pesar de esta igualdad a nivel de funcionalidades, se ha elegido Blender por los siguientes motivos:

- Multiplataforma (3ds Max sólo funciona bajo entorno Windows).
- Bajo consumo de recursos (permite su uso en máquinas con hardware de baja gama).
- Gran comunidad de usuarios (infinidad de material disponible para aprendizaje).
- Gratuito (reduce costes).

El uso de Blender está enfocado al diseño y creación del edificio, así como a los objetos que podremos encontrar. También será necesario para mapear y texturizar los modelos usados.

### *Motor de juego*

El motor de juego definirá la manera de trabajar y el aspecto final del proyecto, por lo que es una decisión importante que debe realizarse tras un exhaustivo análisis de las opciones disponibles.

Aunque los tres motores poseen características y funcionalidades únicas muy atractivas para llevar a cabo un videojuego, el motor elegido ha sido Unity 3D por las siguientes razones:

- Multiplataforma (CryEngine sólo es compatible con Windows).
- Lenguaje de scripting conocido: C# y JavaScript (Unreal y CryEngine usan lenguajes propios).
- Mayor número de plataformas compatibles para exportar: Dispositivos móviles, consolas y PC (Windows, Mac, Linux y navegadores).
- Mayor comunidad de usuarios (infinidad de material disponible para aprendizaje).

Las funciones desarrolladas con Unity 3D consisten en la creación del escenario (empleando todos los diseños y modelos realizados con Blender), así como la interacción entre el usuario y los propios objetos disponibles en la escena.

### *Interacción*

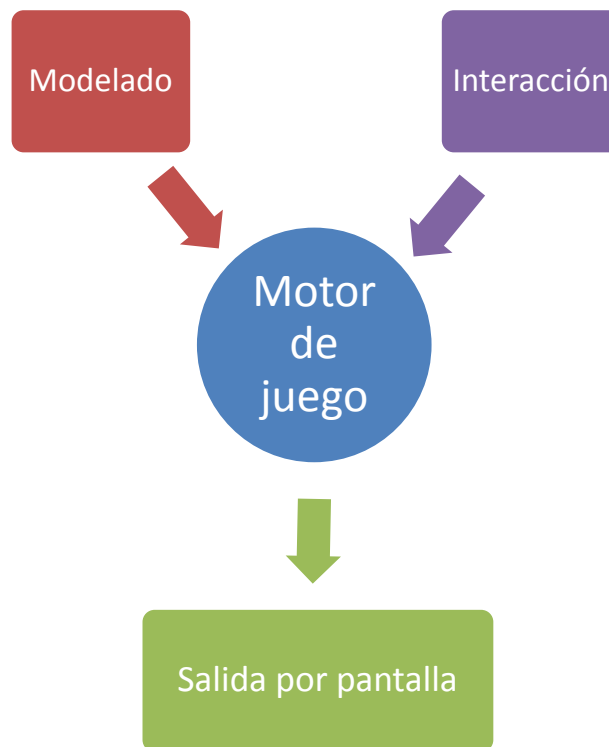
De los métodos disponibles en la actualidad y que se encuentran al alcance de los usuarios a los que va dirigido este videojuego, se ha elegido la combinación de teclado y ratón. El motivo principal es que al ser un desarrollo de videojuego para ordenador, el usuario tendrá que poseer obligatoriamente estos dos periféricos, por lo que también estará habituado a su uso (aunque no sea en entorno de videojuegos, pero deberá saber manejarse con ellos).



No obstante, y gracias a las opciones de interacción que se pueden implementar desde el motor de videojuegos elegido (Unity 3D), en un futuro se podrá añadir la compatibilidad con cualquier otro tipo de dispositivo de interacción que sea necesario, como mandos, pantallas táctiles, detectores de movimiento o gafas de realidad virtual.

#### *Esquema final*

Después de realizar este estudio, queda reflejado que el motor de juego es el encargado de gestionar todas las herramientas y tecnologías que se vayan a usar para poder realizar un producto final completo. En el esquema que se muestra en la *Ilustración 46* se puede observar cuál es el flujo de trabajo gestionado por el motor.



*Ilustración 46: Flujo de trabajo gestionado por el motor de juego*

Después de haber explicado en qué consisten las visitas virtuales, haber realizado un resumen de la historia de la industria de los videojuegos, y analizar las herramientas disponibles para el diseño tridimensional y los sistemas de interacción, en el siguiente apartado vamos a concretar los problemas a los que nos enfrentamos al querer realizar una visita virtual tridimensional dotando a los visitantes de una libertad absoluta a la hora de moverse por dentro del edificio Betancourt de la Universidad Carlos III de Madrid.

## 3. Definición y desarrollo del problema

En el apartado anterior se ha realizado un estudio de los elementos necesarios para poder desarrollar una visita virtual interactiva utilizando las tecnologías de creación de videojuegos, de manera que se ofrezca a los usuarios libertad a la hora de conocer y explorar los entornos en los que está interesado, ya que las opciones disponibles en la actualidad ofrecen una información muy limitada.

Para llevar a cabo la tarea de ofrecer una experiencia mejor y más inmersiva, se va a crear una visita virtual en un entorno conocido como es la Universidad Carlos III de Madrid, concretamente ambientado en el Edificio Agustín de Betancourt, y mediante el uso de las técnicas de movimiento utilizadas en los videojuegos de acción en primera persona, adaptarlo a la experiencia de una visita virtual interactiva.

Por estos motivos, se ha optado por utilizar como base la mecánica de un videojuego FPS (First Person Shooter) en la que la cámara está situada a la altura de los ojos del jugador (vista en 1ª persona), para ayudar a situarse mejor y poder recorrer el edificio en su plenitud.

Este capítulo está dividido en dos apartados importantes: en el primero definiremos el problema al que nos enfrentamos a la hora de realizar el proyecto tanto a nivel de diseño como a nivel de implementación; y en la segunda parte se analizan los requisitos que deberá cumplir la visita virtual, así como la gestión que va a ser necesaria realizar para poder trabajar con toda la información disponible.

### 3.1. Definición del problema

Para poder definir correctamente el problema será necesario realizar un estudio por separado de las diferentes secciones que lo componen, como son el diseño 3D y la implementación en el motor de juego.

#### *Diseño 3D*

Esta es la primera tarea que se deberá realizar, y será llevada a cabo usando el software de modelado Blender. Los modelos y objetos necesarios son:

- Edificio Agustín de Betancourt: composición completa del edificio, tanto externamente como internamente, para permitir una movilidad completa del usuario.
  - Exterior: Paredes, ventanas, puertas, columnas, techos, patios...
  - Interior: Pasillos, puertas, ventanas, luces, escaleras, laboratorios, aulas, talleres, sótano...
- Elementos decorativos como carteles de emergencia, de salida, situación de plantas, planos del edificio...

Todos estos modelos serán también texturizados utilizando Blender, de tal manera que se queden preparados para que el motor gráfico cargue por completo un objeto ya terminado completamente (con forma, tamaño y apariencia correcta).

### *Implementación en el motor de juego*

Cuando los modelos están ya realizados, el siguiente paso es utilizar el motor de juego para unificar todos esos diseños y crear un escenario interactivo donde el usuario pueda explorar el entorno. Este trabajo se realizará mediante Unity 3D, donde el trabajo a realizar será:

- Implementación de los modelos creados anteriormente con Blender.
- Implementación de recursos externos (assets) utilizados para complementar la experiencia.
- Iluminación del entorno.
- Creación de scripts para la interacción del usuario con los objetos, como la apertura de puertas.
- Creación de menús
- Implementación del sistema de interacción usuario-visita virtual mediante el movimiento del personaje y visión en primera persona.

## 3.2. Desarrollo del problema

Esta sección está centrada en el análisis de las características y opciones que deberá cumplir la aplicación mediante:

- Requisitos del sistema y casos de uso.
- Arquitectura de la aplicación y diseño de la visita virtual.
- Gestión de datos y recursos utilizados.

### 3.2.1. Análisis del problema

A través de los requisitos de sistema y los casos de uso, se identificarán las características que deberá cumplir el producto final para considerar que el trabajo realizado ha sido satisfactorio.

#### *Requisitos del sistema*

Los requisitos de sistema están divididos en dos grupos:

- **Requisitos funcionales:** Definen las funciones del software.
- **Requisitos no funcionales:** Definen el diseño y la implementación del software, así como el modo en el que se realizan las tareas.

A continuación se mostrarán dos tablas en las que se enumerarán los distintos tipos de requisitos (tanto funcionales en la *Tabla 3* como no funcionales en la *Tabla 4*), de forma que se pueda obtener de una manera rápida una visión global del contenido. Si se quiere conocer la información completa de cada requisito, podrá ser consultada en su totalidad en el [Anexo III: Requisitos del sistema](#).

Los atributos detallados en las siguientes tablas son:

- **Identificador:** Código único asignado a cada requisito para su identificación.
- **Título:** Nombre breve y descriptivo del requisito.
- **Descripción:** Explicación breve del requisito.

#### Requisitos software funcionales

Identificador	Título	Descripción
RSF-01	Menú Principal	La aplicación deberá poseer un menú principal
RSF-02	Sistema de juego FPS	El tipo de juego corresponderá con un videojuego de género FPS (First Person Shooter)
RSF-03	Escenario	El escenario mostrado debe corresponder con el edificio Agustín de Betancourt
RSF-04	Interacción	El usuario podrá moverse mediante teclado y ratón
RSF-05	Menú Pausa	La aplicación deberá poseer un menú de pausa al que se podrá acceder en cualquier momento de la visita
RSF-06	Calidad Gráfica	La aplicación permitirá cambiar la calidad gráfica durante el transcurso de la visita
RSF-07	Salir Aplicación	La aplicación permitirá salir de la visita en cualquier momento que el usuario lo desee

*Tabla 3: Resumen de requisitos funcionales*

#### Requisitos software no funcionales

Identificador	Título	Descripción
RSNF-01	Multiplataforma	La aplicación deberá ser compatible con distintos S.O.
RSNF-02	Compatibilidad con resoluciones	La aplicación deberá ser compatible con distintas resoluciones
RSNF-03	Fichero ejecutable	La aplicación deberá ser un archivo ejecutable

*Tabla 4: Resumen de requisitos no funcionales*

#### Casos de uso

Los casos de uso son la manera de identificar los pasos o acciones que el usuario deberá realizar para llevar a cabo un proceso en la aplicación.

#### Actores

El proyecto consta únicamente de un actor, que será el visitante. Al tratarse de una aplicación con finalidad informativa y para todos los públicos, el rol de actor lo puede desarrollar cualquier tipo de persona, independientemente del género o la edad, ya que el objetivo es dar a conocer el entorno universitario a cualquier posible visitante.

Siguiendo la misma estructura que con los requisitos de sistema, en la *Tabla 5* se enumerarán los casos de uso existentes en la aplicación, para conocer rápidamente las distintas interacciones entre el visitante y las opciones presentes en el videojuego. Si se desea ampliar la información y conocer mejor cada caso de uso, la información podrá ser consultada en el [Anexo IV: Casos de uso](#).

Los atributos detallados en la tabla de casos de uso son:

- **Identificador:** Código único asignado a cada caso de uso.
- **Título:** Nombre breve y descriptivo del caso de uso.
- **Descripción:** Explicación breve del caso de uso.

Identificador	Título	Descripción
CU-01	Visita Virtual	El visitante tiene como opción acceder a la visita virtual
CU-02	Salir desde Menú Principal	El visitante tiene como opción salir de la aplicación
CU-03	Mover cámara	El visitante puede mover la cámara moviendo el ratón
CU-04	Mover personaje	El visitante puede mover al personaje usando las teclas de dirección o las teclas W, A, S, D del teclado
CU-05	Abrir/Cerrar Puerta	El visitante podrá abrir/cerrar determinadas puertas pulsando la “tecla E”
CU-06	Menú Principal	Desde el menú de pausa, el visitante podrá volver a la pantalla de inicio
CU-07	Cambiar Calidad Gráfica	Desde el menú de pausa, el visitante podrá modificar la calidad gráfica de la visita virtual
CU-08	Salir desde Menú Pausa	Desde el menú de pausa, el visitante podrá salir de la aplicación

*Tabla 5: Resumen de casos de uso*

### *Relación entre Requisitos del sistema y Casos de uso*

A continuación se va a mostrar en la *Tabla 6* la relación existente entre los requisitos funcionales y los casos de uso, de manera que se pueda apreciar rápidamente la conexión entre ellos, como en el caso del RSF-07 “Salir de la Aplicación”, que está disponible desde el CU-02 “Salir desde el Menú Principal” y desde el CU-08 “Salir desde el Menú Pausa”.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08
RSF-01	X	X				X		
RSF-02			X	X	X			
RSF-03			X	X	X		X	
RSF-04			X	X	X			
RSF-05						X	X	X
RSF-06							X	
RSF-07		X						X

*Tabla 6: Relación entre Requisitos del sistema y Casos de uso*

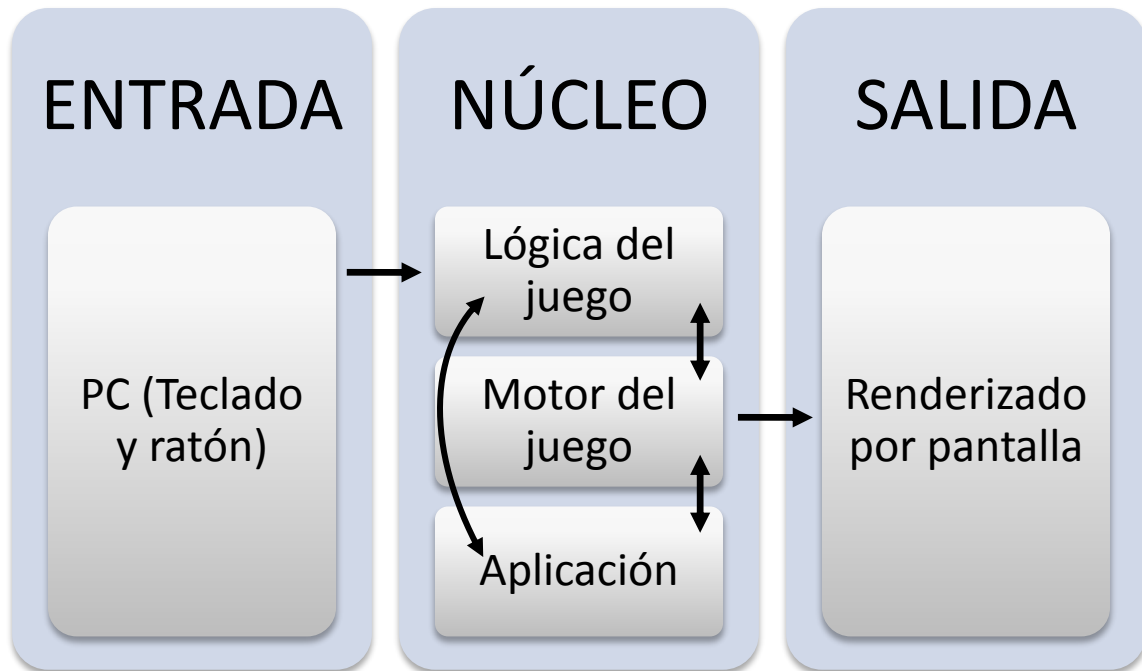
### **3.2.2. Diseño de la solución**

En esta sección del capítulo se pondrá en conocimiento los pasos seguidos para el diseño de la aplicación desarrollada. El videojuego consiste en una visita virtual al edificio Agustín de Betancourt del campus de Leganés, por lo que el usuario podrá desplazarse libremente por el edificio, como si fuese un visitante o alumno que acude al edificio.

### Arquitectura de la aplicación

La arquitectura de la aplicación nos permite obtener un diseño a alto nivel, de tal manera que se pueda identificar en el sistema las partes y módulos que lo componen, y la relación existente entre ellos.

Para facilitar la comprensión, se mostrará en el esquema de la *Ilustración 47* las relaciones existentes en la arquitectura, y posteriormente se realizará una explicación de cada parte.



*Ilustración 47: Esquema de la arquitectura de la aplicación*

**Entrada:** Parte encargada de capturar las acciones realizadas por el usuario.

- **PC (Teclado y ratón):** Captura las indicaciones que el usuario quiere ver reflejadas en la aplicación.

**Núcleo:** Elemento de mayor importancia en la aplicación, ya que abarca la lógica de juego, el motor y la aplicación.

- **Motor de juego:** Encargado de controlar los distintos parámetros y aspectos de la aplicación, como es el movimiento del personaje, iluminación, físicas, colisiones, animaciones, entorno, control de datos, renderizado de objetos y salida por pantalla (render).
- **Lógica del juego:** Código de programación que establece las conexiones entre los elementos de la aplicación y el comportamiento del personaje. Además, interpreta los datos que llegan desde los periféricos de entrada para convertirlos en las acciones que el usuario quiere ver representado en la visita virtual.



- **Aplicación:** Elemento principal del proyecto, encargado de unificar los componentes de mayor importancia y de ejecutar la aplicación. También se encarga de controlar el flujo de datos, el motor y la lógica de juego.

**Salida:** Parte encargada de mostrar al visitante el renderizado generado por el juego.

- **Renderizado por pantalla:** Salida del renderizado del juego para que el visitante pueda observar qué ocurre durante la visita.

La modificación de la capa de entrada sólo afectaría a la manera de interactuar entre el usuario y la aplicación (usando un mando, o controles de movimiento, por poner un ejemplo), mientras que modificar la capa de salida se vería reflejado en la manera en la que la visita es mostrada al usuario (por ejemplo, mediante el uso de gafas de realidad virtual, en vez de por el monitor del ordenador). Sin embargo, la modificación del núcleo afecta directamente al producto final, ya que engloba la manera en la que se trabaja con los objetos o componentes como físicas e iluminación, o la lógica del juego.

### **Modelado del escenario**

El modelado del escenario ha sido realizado íntegramente con la herramienta Blender partiendo desde cero y sin utilizar ningún modelo u objeto ya disponible en la red.

Se ha decidido realizar un edificio de la Escuela Politécnica Superior del campus de Leganés ya que se han cursado los estudios en dicho entorno de la Universidad Carlos III de Madrid. En concreto, se ha elegido el edificio Agustín de Betancourt por ser el edificio más grande y con mayor variedad de entornos de los construidos en el campus.

Existen otros edificios, como es el ed. Juan Benet, que se reduce a un edificio de varias plantas compuesto por un único pasillo, o el ed. Sabatini que es un "cuadrado" con un patio interno de tres plantas que repite la misma estructura en sus diferentes alturas, mientras que el ed. Betancourt posee cinco plantas (desde el parking situado en el sótano hasta la tercera planta) con diferente estructura cada una, en las que hay multitud de despachos, laboratorios, aulas de teoría y ordenadores, talleres, salones de conferencias, varios patios, tiendas, parking... En total, posee más de 200 metros de largo y 75 metros de ancho, que lo convierte en el edificio más grande y más completo del campus.

Para poder llevar a cabo la planificación del modelo del edificio se buscó multitud de información para conseguir que nuestro modelado fuese lo más realista posible.

Por un lado se partía del conocimiento adquirido tras varios años de estudios realizados entre las paredes del edificio Betancourt, pero las medidas que se pudiesen conseguir de la experiencia personal podrían distar notablemente de la realidad, por lo que se optó por realizar un amplio trabajo de documentación fotográfica en el que se realizaron más de 650 fotografías.

A pesar de estas muestras gráficas, y al querer lograr un acabado realista para poder conseguir que los visitantes obtuviesen una experiencia lo más cercana a la realidad, se buscaron los planos oficiales en Internet. En la página web de la Universidad, la Unidad Técnica de Mantenimiento y

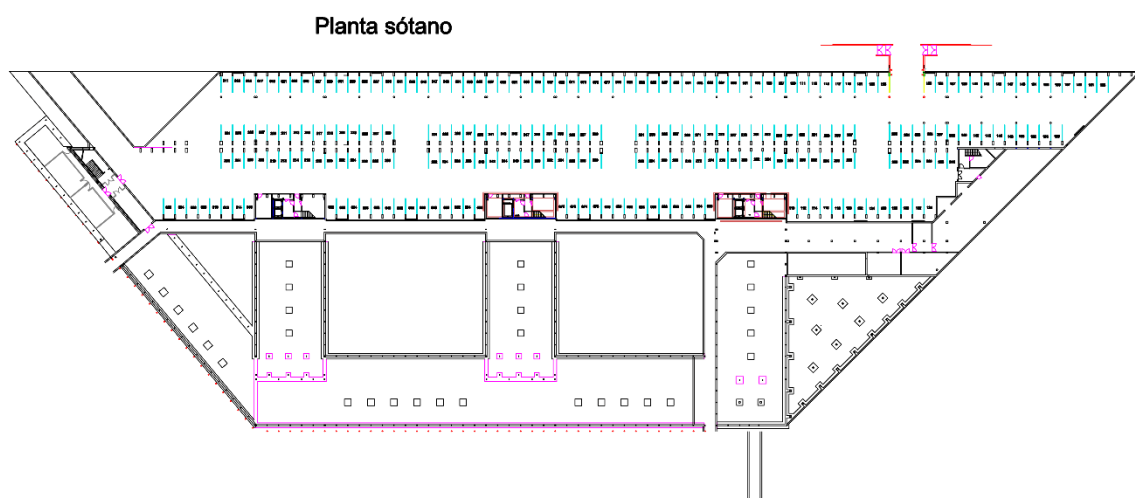
Obras tiene a disposición de cualquier visitante (no es necesario ser miembro de la Universidad) los planos técnicos de todos los campus. Los planos referentes al edificio Betancourt pertenecen a las últimas obras realizadas en 2004, por lo que es posible que exista cierto desfase en algunos laboratorios o despachos que hayan realizado pequeñas modificaciones posteriores a la publicación del material de apoyo utilizado. [19]

Aun así, las medidas proporcionadas por estos planos son parciales (viene la distribución de las salas y el área ocupada por cada una, pero no el espacio total que ocupa el edificio), por lo que se utilizó la herramienta de medición de distancias de Google Maps para calcular la distancia total y saber qué superficie ocupa el edificio [20]. A partir de ese momento, se puede empezar a realizar el modelado del edificio en todas sus plantas.

Por último, y para terminar de poder construir el edificio, nos encontramos con que no tenemos ninguna referencia sobre las medidas verticales. Los planos nos ofrecen una representación bidimensional completa, pero para la dimensión relativa a la altura de los objetos necesitamos documentación que no viene en los planos. Para suplir esta carencia de información, se ha buscado en el CTE (Código Técnico de la Edificación) [21] las medidas que deben cumplir las edificaciones en relación con los requisitos básicos de seguridad y habitabilidad establecidos por el Ministerio de Fomento y que deben cumplir todas las construcciones.

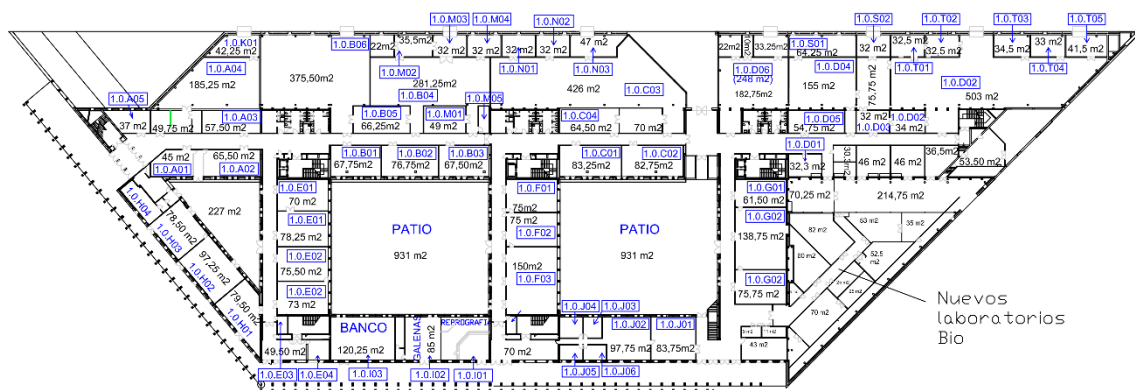
De esta manera, y usando entre otros, el Documento Básico de Seguridad de Utilización y Accesibilidad (DB-SUA) y el Documento Básico de Seguridad en caso de Incendio (DB-SI), se han conseguido todas las medidas necesarias para situar los techos, ventanas y barandillas a la altura correspondiente, o saber las medidas exactas que deben tener las distintas puertas, los peldaños de las escaleras o el espacio que ocupan las entreplantas.

A continuación se muestran los distintos planos técnicos obtenidos (*Ilustración 48, Ilustración 49, Ilustración 50, Ilustración 51 e Ilustración 52*), así como diferentes fotografías tomadas del edificio (*Ilustración 53, Ilustración 54, Ilustración 55 e Ilustración 56*), para poder llevar a cabo el modelado.



*Ilustración 48: Ed. Betancourt. Plano técnico de la "Planta Sótano"*

## PLANTA BAJA



UNIVERSIDAD CARLOS III DE MADRID, CAMPUS DE LEGANÉS

EDIFICIO BETANCOURT

Escala 1:600

Enero de 2004

Ilustración 49: Ed. Betancourt. Plano técnico de la "Planta Baja"

## Planta primera ESCALA 1:600



UNIVERSIDAD CARLOS III DE MADRID, CAMPUS DE LEGANÉS

EDIFICIO BETANCOURT

Escala 1:600

Enero de 2004

Ilustración 50: Ed. Betancourt. Plano técnico de la "Planta Primera"

## Planta segunda

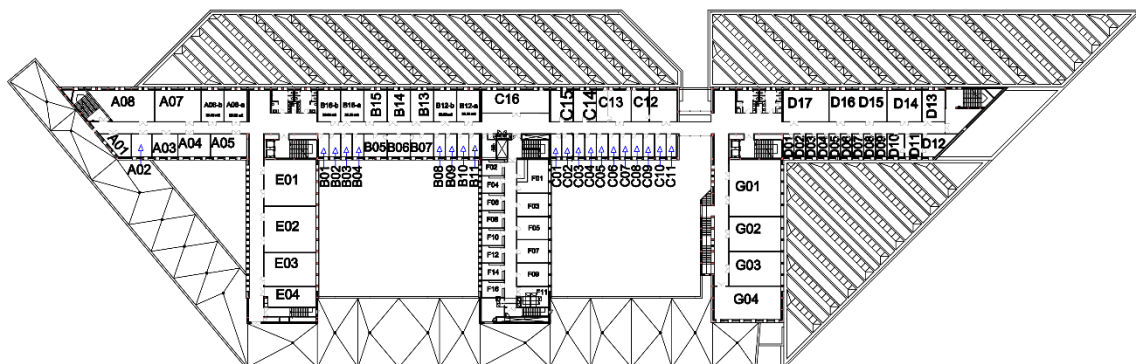
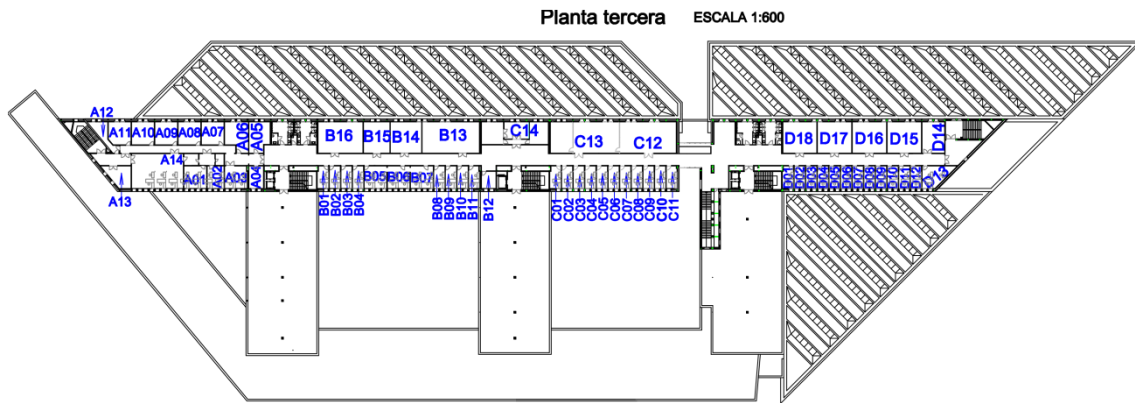


Ilustración 51: Ed. Betancourt. Plano técnico de la "Planta Segunda"



*Ilustración 52: Ed. Betancourt. Plano técnico de la "Planta Tercera"*



*Ilustración 53: Vista exterior del Ed. Betancourt desde el ed. Sabatini  
Imagen realizada para la documentación fotográfica*

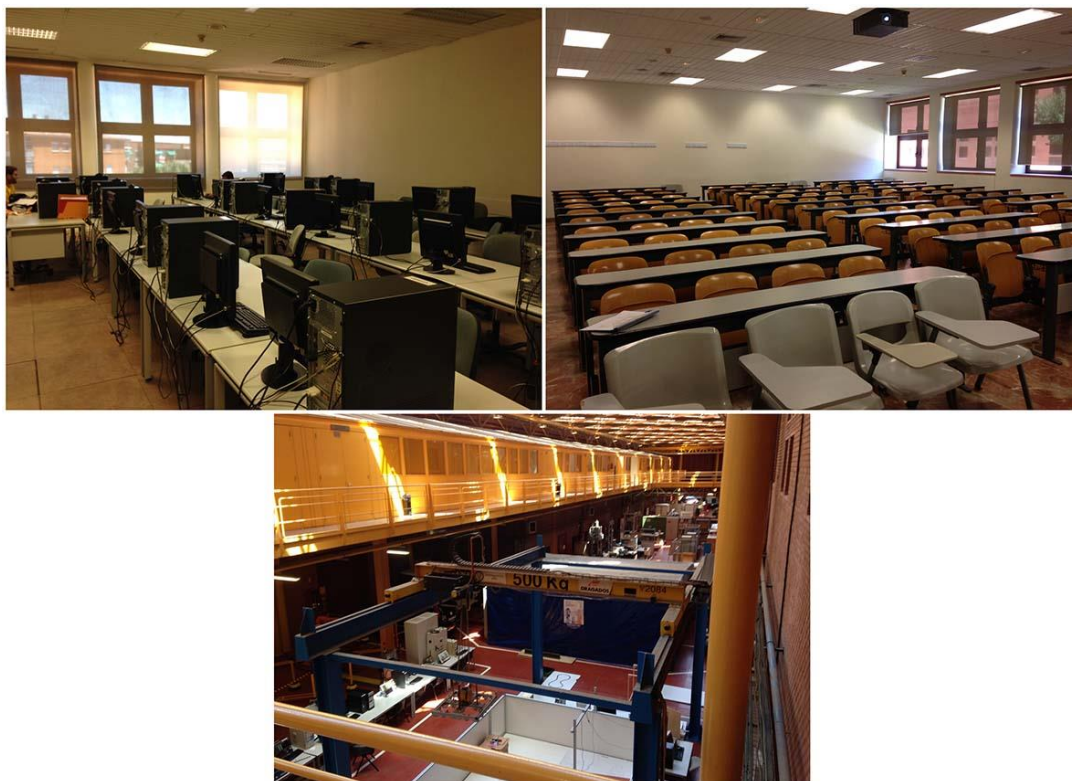


*Ilustración 54: Vista exterior del Ed. Betancourt desde la cúpula de la biblioteca  
Imagen extraída de la "Visita Virtual" de la web de la UC3M*





*Ilustración 55: Ed. Betancourt. Vista interior de los pasillos  
Imágenes realizadas para la documentación fotográfica*



*Ilustración 56: Ed. Betancourt. Vista interior de aulas y laboratorios  
Imágenes realizadas para la documentación fotográfica*

## Menús

Los menús son componentes de la aplicación que son necesarios implementar mediante el motor de juego, por lo que en esta sección simplemente se mostrarán y explicarán los bocetos realizados, de tal manera que sirvan como guía cuando se les dé forma en Unity 3D.

En los menús que se van a realizar se busca dar un aspecto minimalista, de tal manera que la información que se quiere mostrar sea clara y exista el menor número de distracciones. El resultado final se podrá observar más adelante en la sección “4.2.6. Menús”.

### Menú Principal

El menú principal se mostrará nada más abrir la aplicación, y las únicas opciones posibles serán la del acceso a la visita virtual y la de salir de la aplicación. El boceto inicial se puede observar en la *Ilustración 57*.



*Ilustración 57: Boceto "Menú Principal"*

### Menú de Pausa

Durante la visita, el usuario podrá acceder al menú de pausa pulsando la “tecla ESC”, y se le deberán dar varias opciones: volver a la pantalla inicial, salir directamente de la aplicación, o cambiar la calidad gráfica de la visita virtual. La *Ilustración 58* muestra el esquema visual que deberá seguir el menú.

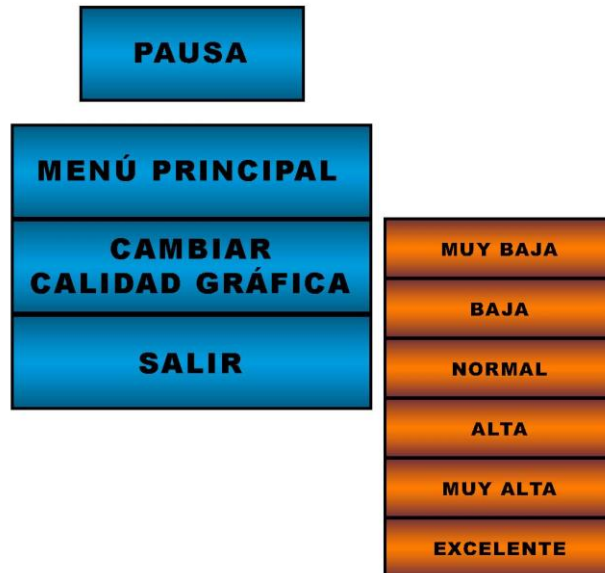


*Ilustración 58: Boceto "Menú Pausa"*



### Menú Cambiar Calidad Gráfica

Si durante la pausa de la visita virtual el usuario considera que las opciones de calidad gráfica no son las idóneas para él, podrá cambiarlas accediendo al submenú de "Cambiar Calidad Gráfica", donde estarán disponibles diferentes calidades, según se muestra en la *Ilustración 59*.



*Ilustración 59: Boceto "Menú Cambiar Calidad Gráfica"*

### 3.2.3. Gestión de datos

En este apartado se describirán los formatos utilizados para la exportación e importación de los archivos necesarios para trabajar. De la misma manera, también se explicará cómo se han gestionado los datos, tanto en Blender para la creación de modelos, como en Unity para conseguir que el motor trabajase de manera óptima con los recursos con los que se contaba.

#### *Formatos de exportación/importación*

Los siguientes formatos explicados corresponden a todos los tipos distintos de archivos que se han utilizado en la realización del proyecto.

**3DS:** formato de archivo usado por algunos programas de simulación y gráficos 3D (en este proyecto, se usa con el programa 3ds Max).

**BLEND:** formato utilizado por Blender para guardar los proyectos creados con este software.

**CS:** extensión de archivo que contiene código en lenguaje de programación C#. Utilizado para crear scripts que se usarán en Unity 3D.

**FBX:** formato de archivo 3D independiente de la plataforma, que proporciona acceso al contenido creado en cualquier paquete de software. Suele ser el formato de exportación por defecto de los modelos cuando se quiere poder trabajar con ellos en distintas herramientas.

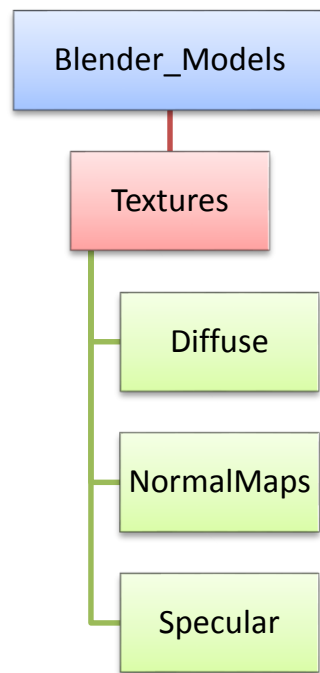
**JPG/JPEG:** siglas de Joint Photographic Experts Group, es un formato gráfico que utiliza un algoritmo de compresión con pérdida para reducir el tamaño de los archivos de imágenes.

**JS:** extensión de archivo que contiene código en lenguaje de programación JavaScript. Utilizado para crear scripts que se usarán en Unity 3D.

**PNG:** siglas de Portable Network Graphics, es un formato gráfico para guardar imágenes basado en un algoritmo de compresión sin pérdida de bitmaps.

### *Gestión de datos en Blender*

Para generar los modelos en Blender se ha utilizado optado por usar la jerarquía de carpetas que se muestra en la *Ilustración 60*:



*Ilustración 60: Jerarquía de carpetas para Blender*

Carpeta **Blender\_Models**: Contiene todos los modelos creados mediante Blender y la carpeta texturas.

Carpeta **Textures**: Contiene los tres tipos de texturas utilizados en la generación de los modelos.

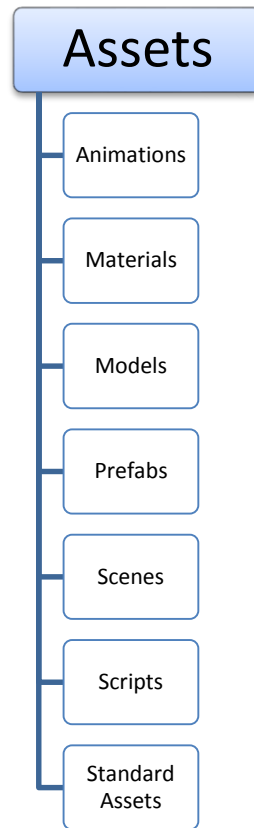
Carpeta **Diffuse**: Contiene las imágenes con la información básica de la textura, como el color y las formas.

Carpeta **NormalMaps**: Contiene la información de la imagen relativa al ángulo de incidencia de la luz sobre cada parte de la textura. Es utilizada para dar efecto de relieve y rugosidad a las texturas.

Carpeta **Specular**: Contiene la información de la imagen relativa al brillo que posee la textura. Es utilizada para indicar en qué zonas el reflejo de la iluminación debe ser mayor o menor.

### Gestión de datos en Unity 3D

Al crear un proyecto, Unity 3D crea automáticamente una jerarquía básica de carpetas para ayudar a tener ordenados todos los componentes del proyecto. Este directorio, llamado Assets, puede ser modificado según sea necesario. En este caso, se ha respetado la estructura por defecto creada por el motor, añadiendo las carpetas necesarias para ordenados toda la información necesaria para realizar el proyecto, como queda reflejado en la *Ilustración 61*.



*Ilustración 61: Jerarquía de carpetas para Unity 3D*

Carpeta **Animations**: Contiene las animaciones que serán usadas por los objetos.

Carpeta **Materials**: Contiene los materiales usados por los distintos objetos y escenas del juego.

Carpeta **Models**: Contiene todos los modelos usados para recrear la visita. Dentro de esta carpeta se encuentra la estructura de los objetos generados con Blender, mostrada en la sección anterior "Gestión de datos en Blender".

Carpeta **Prefabs**: Contiene la información de los objetos "prefabs". Estos modelos son objetos con unas determinadas propiedades y actúan como una plantilla, a partir de la cual se pueden crear nuevas instancias del objeto en la escena. Cualquier edición hecha a un prefab será inmediatamente realizada en todas las instancias producidas por él.

Carpeta **Scenes**: Contiene las escenas, niveles o menús del juego. En nuestro caso, la "escena" menú principal y la "escena" de la visita virtual.

Carpeta **Scripts**: Contiene el código necesario para hacer funcionar la lógica del juego.

Carpeta **Standard Assets**: Contiene los assets que trae por defecto Unity 3D y que han sido utilizados, como el controlador del personaje o el skybox (cielo).

Tras hablar en este capítulo del problema al que nos enfrentamos y la tarea que queremos desarrollar (realizar una visita virtual interactiva en la que el usuario tenga libertad de movimiento por cualquier parte del edificio), se han analizado las características y opciones que la aplicación debe cumplir, justificando la información utilizada para su estudio y el motivo para considerarse relevante, además de explicar la arquitectura interna y la gestión de los recursos disponibles.

La siguiente sección está destinada a explicar ampliamente y con detalle todo el proceso seguido para crear la visita virtual, desde la creación de modelos tridimensionales con Blender, hasta la implementación mediante Unity 3D para conseguir que los visitantes puedan moverse e interactuar con el edificio diseñado.

## 4. Implementación

En la sección anterior se han descrito las características que deberá cumplir el proyecto para considerar que el trabajo realizado ha sido completo. Dentro de estas características se encuentran los requisitos del sistema, casos de uso, arquitectura de la aplicación, diseño de los componentes de la visita y la gestión de datos.

En este capítulo, dedicado a la implementación de la solución, se explicará con detalle todo el proceso de creación de cada parte del proyecto, así como las tecnologías utilizadas para poder dar el aspecto deseado al producto final. En la *Ilustración 62* se muestran distintas zonas del edificio Betancourt con su acabado final, de tal manera que cuando se expliquen en el proceso de creación se pueda apreciar con claridad cada componente creado.



*Ilustración 62: Resultado final de la visita virtual*

### 4.1. Modelado

El modelado, como se ha comentado anteriormente, se ha realizado con el programa Blender. La versión distribuida al inicio del proyecto era la v2.70, por lo que ha sido la versión principal en la que se han realizado todos los modelados.

Los modelos se han mantenido con la extensión .blend, ya que Unity 3D es capaz de detectar cualquier cambio realizado en el modelo original y re-importar automáticamente el objeto. Esta característica es muy útil por si se detecta algún fallo en el objeto original o simplemente se quieren realizar modificaciones, ya que al instante obtendremos el modelo actualizado en el motor sin necesidad de exportar el archivo desde Blender y sin tener que indicar a Unity 3D que debe re-importar el archivo.

#### 4.1.1. Estructura del edificio Agustín de Betancourt

El edificio Agustín de Betancourt será el escenario principal en el que se desarrollará la visita, por lo que el trabajo realizado ha sido máximo para conseguir la misma apariencia que la del edificio real, de tal manera que el usuario sienta que se encuentra verdaderamente visitando el edificio.

Para modelar el edificio se ha dividido la tarea en trabajo interior y trabajo exterior. A su vez, el trabajo del interior del edificio se ha repartido por plantas, de tal manera que se pueda llevar el control del trabajo más fácilmente. Además, después de terminar el armazón del edificio, se ha trabajado en la decoración interior para terminar de completar el edificio.

#### Edificio Betancourt – Interior

Nada más comenzar el diseño nos encontramos con un problema importante. Los documentos obtenidos del edificio no mantienen las mismas proporciones en las diferentes plantas. Esto es debido a que al ser suministrados todos en .pdf en formato DIN A-4, las medidas han debido ser adaptadas al espacio disponible. En los documentos se indica que la escala es 1:600, sin embargo, al compararlos entre sí se ve que el tamaño es distinto en cada uno, como se puede apreciar en los planos mostrados anteriormente en la sección “Modelado del escenario”, en la *Ilustración 48*, *Ilustración 49*, *Ilustración 50*, *Ilustración 51* e *Ilustración 52*, correspondientes a las distintas plantas que componen el edificio.

Por este motivo, el primer paso a realizar ha sido calcular el tamaño del edificio, para lo cual se utilizó la herramienta de medición que ofrece Google Maps y así tomar varias medidas (*Ilustración 63*) que luego servirán para situar el plano en Blender con las proporciones reales, de tal manera que todo lo construido a partir de los planos conseguidos mantenga unas dimensiones realistas.

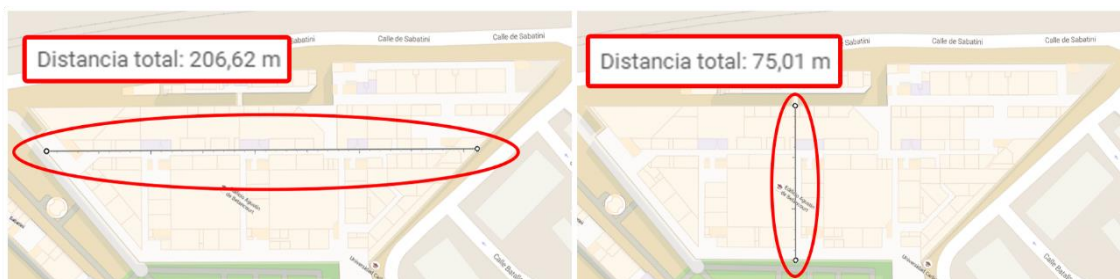


Ilustración 63: Mediciones ed. Betancourt

Uso de la herramienta de cálculo de mediciones de Google Maps para medir el ed. Betancourt.



Después de tener diferentes medidas de referencia, se añadieron en Blender los planos oficiales conseguidos en la web de la Universidad, de esta manera el trabajo realizado sería fiel a la realidad, al trabajar con las dimensiones exactas con las que se construyó el edificio. A partir de ese momento, el trabajo se realiza manteniendo las proporciones y medidas que se indican en los planos de las distintas plantas, lo que ayudará tanto a la hora de crear el modelo, como para conseguir realismo.

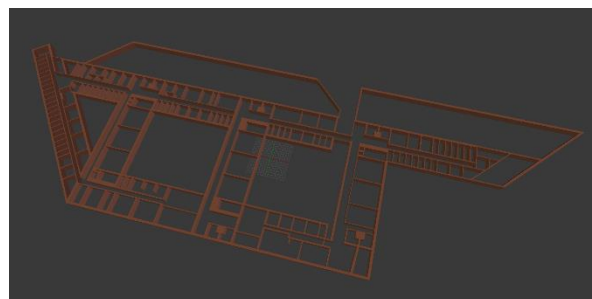
El primer paso es, sobre el plano de cada suelo, crear todas las paredes, de tal manera que consigamos un "esqueleto" de la planta. Es muy importante tener en cuenta que cada planta posee una estructura interna diferente, por lo que a excepción de las paredes externas y las estancias comunes (como los talleres que ocupan dos plantas, o las escaleras que comunican todas las alturas del edificio), no se puede copiar el modelo de una planta a otra directamente, siendo necesaria su creación desde cero (o en el caso de querer aprovechar el modelo de otra planta por las paredes exteriores o zonas comunes, serán necesarias grandes modificaciones).



**Ilustración 64: Creación de paredes, vista planta**  
Se pueden observar las paredes creadas (resaltadas en naranja) sobre el plano original, comprobando que existe una correspondencia exacta.

Como se puede observar en la *Ilustración 64*, al crear las paredes (señaladas en naranja) teniendo como guía el plano original, la correspondencia entre el modelado y el diseño original es exacto, por lo que el usuario se va a encontrar con una representación fiel a la construcción realizada.

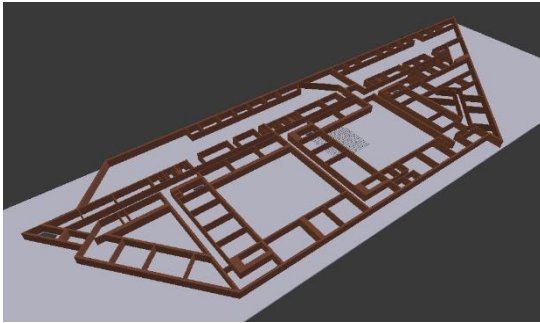
Al llegar a este punto existe otro problema. Todos los planos obtenidos son en proyección planta, proporcionando información sólo dos dimensiones, por lo que no se posee ningún tipo de referencia para crear la altura de los objetos y poder crear nuestra vista tridimensional. Como hemos comentado anteriormente en el apartado “*Modelado del escenario*”, se ha buscado la información relativa a las alturas en el CTE, y proporcionar las alturas exactas a todos los elementos que compondrán el modelado final del edificio Betancourt.



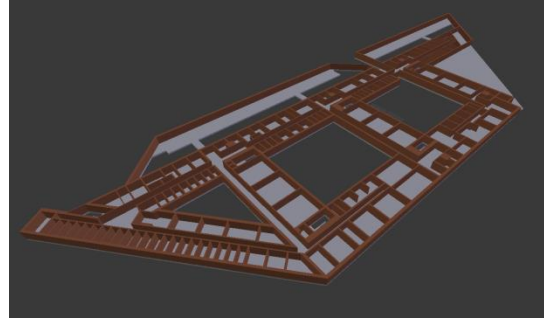
**Ilustración 65: Creación de paredes, vista en perspectiva**  
Se muestra el resultado final de la creación de las paredes de una de las plantas, tras haber sido tomadas las medidas sobre el plano para obtener una correspondencia exacta.

En la *Ilustración 65* se puede ver como se le ha dotado de altura a las paredes diseñadas anteriormente, de tal manera que poseamos un primer modelado tridimensional del armazón que compone una de las plantas del edificio.

Tras tener hechas todas las paredes sobre un plano cuadrado que hace las funciones de suelo, es necesario darle forma al piso. Esto es primordial para poder dejar libres las vías de acceso entre plantas por las escaleras, o no tapar los patios o talleres, así como tener el aspecto exterior deseado (*Ilustración 66*, *Ilustración 67*).

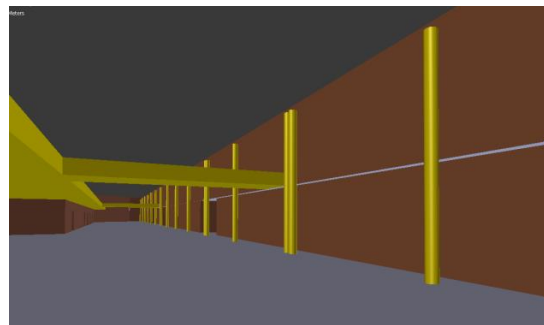


*Ilustración 66: Planta baja con el suelo sin cortes*  
Vista de la planta baja con el suelo y las paredes modeladas en su forma final.



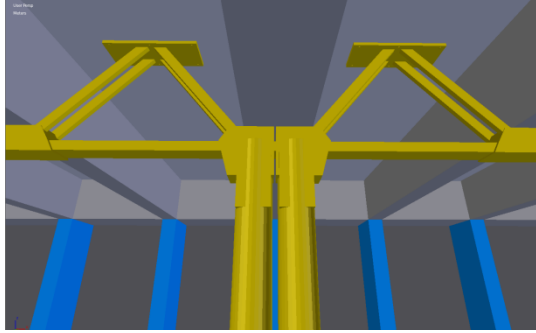
*Ilustración 67: Planta primera con el suelo modelado*  
Vista de la primera planta con las paredes modeladas y el suelo editado con la forma del edificio, los talleres y los patios.

El siguiente paso para crear el "esqueleto" de nuestro edificio serán las columnas y vigas que nos encontramos a lo largo de las diferentes estancias. En el sótano, las plazas de garaje están separadas por columnas, las cuales están unidas entre ellas por vigas. En la planta baja, más concretamente en los talleres, encontramos columnas alrededor de la estancia que sirven como soporte para el techo (*Ilustración 68*).

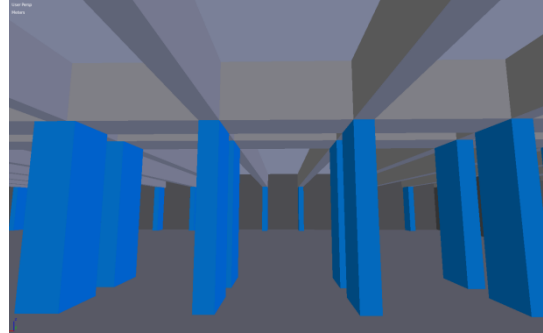


*Ilustración 68: Columnas de los talleres de la planta baja*  
Detalle de las columnas redondas que componen la estructura de los talleres.

Las columnas y vigas del sótano están compuestas de dos tipos distintos de estructuras. Por un lado tenemos el patrón principal de columnas y vigas cuadradas, las cuales componen la gran mayoría de columnas y separadores de plazas de aparcamiento (*Ilustración 70*), y por otro están las columnas y vigas "amarillas", que están situadas a en un lateral del parking, pero que poseen una estructura muy particular (*Ilustración 69*). Ambos tipos de estructuras han sido modelados con sus características propias, para que puedan ser reconocidos rápidamente.



**Ilustración 69: Columnas y vigas "amarillas" del sótano**  
Detalle de las columnas, vigas y soportes del techo que componen la estructura auxiliar "amarilla" situada en el parking del sótano.



**Ilustración 70: Columnas y vigas cuadradas del sótano**  
Detalle de las columnas y vigas cuadradas que componen la estructura principal del parking situada en el sótano.

Es importante remarcar que a las columnas redondas (y a todas las figuras con superficies curvas que se vayan creando en adelante) se les va a aplicar un acabado "suavizado" (smooth) para que la superficie se aprecie redondeada. Esta acción es necesaria para que no se aprecie que es un prisma de múltiples caras y se obtenga mejor resultado. En la *Ilustración 71* se observa la diferencia entre la columna de la izquierda (con el suavizado desactivado y múltiples superficies planas) y la columna de la derecha (con el suavizado activado y una superficie redondeada).



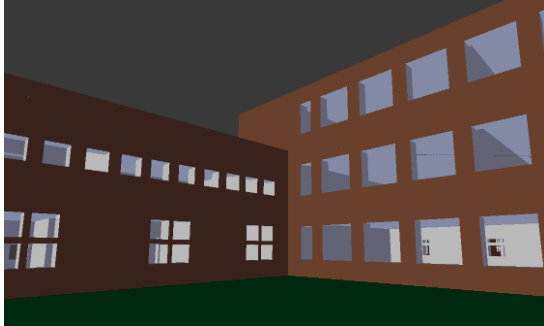
**Ilustración 71: Acabado de columna con y sin suavizado**  
Diferencia entre un objeto con caras "planas" a la izquierda y caras "suavizadas" a la derecha.

En este punto se puede considerar que ya tenemos el almacén principal del edificio modelado, lo que incluye suelos, paredes, columnas y vigas. A pesar de que parezca un trabajo rápido y sencillo, la realidad es completamente opuesta. Esta parte del modelado es la que más tiempo conlleva debido a la complejidad en la estructura, distribución de las aulas y múltiples plantas a realizar.

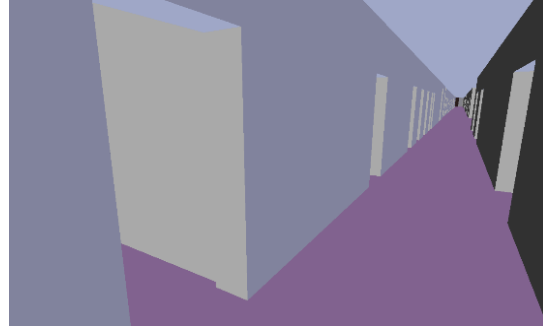
El siguiente paso será crear los huecos para las puertas y las ventanas. Como se ha explicado anteriormente, mediante la información extraída del "Código Técnico de la Edificación" (CTE) se ha podido calcular la altura que deberían tener las puertas, o la distancia a la que deben estar situadas las ventanas del suelo, ya que de los documentos de apoyo conseguidos no se obtiene ninguna información referente al plano vertical.

Tras conseguir las múltiples medidas referentes a los distintos tipos de ventanas y puertas existentes (puertas de paso, de emergencia, de salida, cortafuegos...), se ha procedido a realizar el "vaciado" correspondiente en las paredes ya creadas, para ir dotando poco a poco del resultado final que necesitaremos reflejar cuando el modelado se encuentre terminado.

En la *Ilustración 73* se puede observar en uno de los pasillos principales cómo se han realizado los huecos en las paredes correspondientes a las puertas, para que más adelante, cuando estén diseñadas y modeladas cada una de las distintas puertas, puedan ser colocadas en su lugar correspondiente. Realizando la misma técnica, en la *Ilustración 72* se puede observar el resultado final de las paredes tras haber realizado el vaciado correspondiente a los distintos tipos de ventanas.



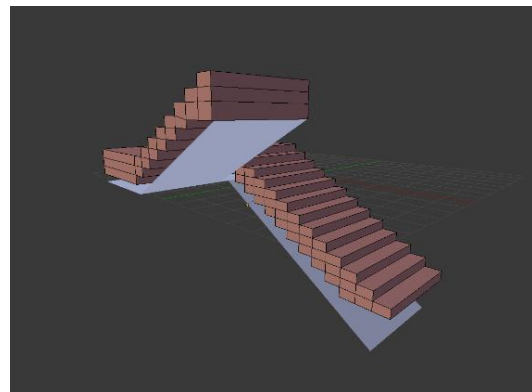
*Ilustración 72: Detalle de los huecos para las ventanas*  
Detalle de las paredes de uno de los patios, en donde se pueden observar los huecos realizados para las ventanas.



*Ilustración 73: Detalle de los huecos para las puertas*  
Detalle de las paredes de uno de los pasillos principales, donde se puede observar los huecos realizados para las puertas.

Después de haber creado los suelos, las columnas y las paredes, y de haber preparado éstas últimas para la colocación posterior de puertas y ventanas realizando los huecos necesarios, sólo nos faltaría crear las escaleras para tener un modelado interno completamente funcional a través del cual nos podríamos mover con total libertad, aunque carecería completamente de texturas.

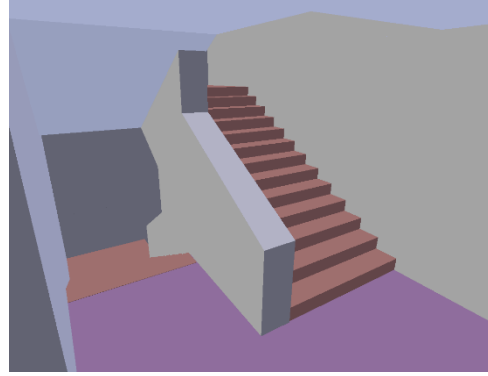
Para la realización de las escaleras, de la misma manera que con las puertas y las ventanas, se ha buscado en los documentos del CTE las medidas que deben cumplir. Tanto los peldaños (compuestos por huella y contrahuella), los rellanos, el ancho mínimo de paso y la altura del pasamanos han sido tenidos en cuenta para que las proporciones fuesen realistas. El resultado, a falta únicamente del pasamano, puede verse en la *Ilustración 74* junto a una “cubierta” posicionada en la parte inferior, que a la vez que evita que se vea el modelado, hace las funciones de techo en la planta inmediatamente inferior.



*Ilustración 74: Detalle de la creación de las escaleras*  
Se puede ver en detalle cómo se han creado las escaleras junto con la cubierta inferior que realiza funciones de techo.

Para acabar el modelado de las escaleras, y a la vez el almacén general del edificio, sólo faltaría colocar el pasamano interior, que es común a toda la sección de escaleras desde el sótano hasta la tercera planta, por lo que se ha creado de una sola pieza.

En la *Ilustración 75* se puede observar el resultado final de todo el conjunto, pudiéndose ver las escaleras, junto con la cubierta de techo (en este caso correspondiente a las escaleras de la planta superior), y el pasamano central común a todos los tramos de escaleras.



*Ilustración 75: Resultado final de las escaleras*  
Se puede ver el resultado final de las escaleras, con la cubierta en la parte superior y el pasamano común a todos los tramos de escaleras.

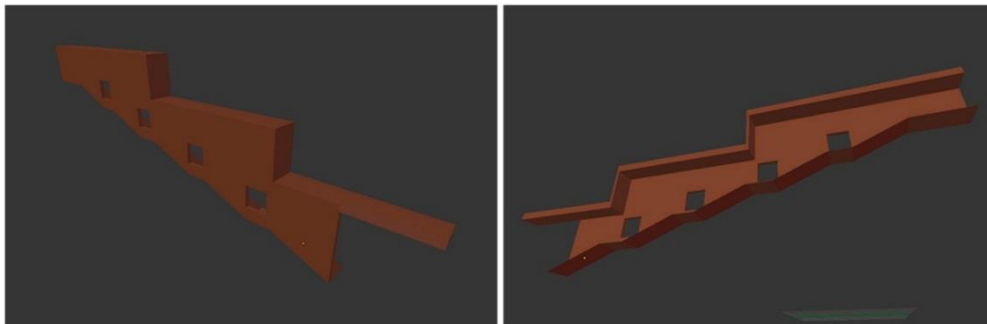
### **Edificio Betancourt – Exterior**

Después de tener realizada la estructura interna del edificio, el siguiente paso es detallar el aspecto exterior que caracteriza al edificio Betancourt. Para poder llevar a cabo esta tarea, aparte de usar los planos técnicos (como en la sección interior), también se han utilizado diferentes fotografías tomadas del exterior del edificio y las imágenes aéreas disponibles en Google Maps (para poder observar las azoteas).

Varios elementos realizados en esta etapa se les puede considerar innecesarios a la hora de realizar una visita por el interior del edificio, pero son estos detalles los que ofrecen una sensación completa de realismo al entorno visitado cuando realicemos nuestro paseo interactivo.

El primer trabajo a realizar es terminar de “cerrar” las paredes, ya que para realizar la escalera del “pasillo G”, denominada a partir de ahora escalera “principal” (la cual empieza en conserjería y sube hasta la tercera planta), de los planos sólo se puede extraer la información referente al ancho de la escalera y las zonas de unión con cada planta, pero nada que ayude a desarrollar externamente su aspecto.

Para conseguir el aspecto externo real, se han realizado fotografías desde el patio y desde las aulas del “pasillo F” (situado al otro lado del patio) que sirviesen como guía. En la *Ilustración 76* se observa el resultado obtenido, en donde se puede apreciar todo el trabajo realizado.



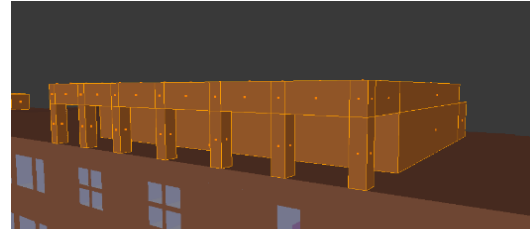
*Ilustración 76: Detalle de la escalera principal*  
Vista del exterior (izquierda) y del interior (derecha) del modelado de la escalera principal.

Siguiendo con el trabajo de exteriores, también se han construido las salas de maquinaria de los ascensores. Aunque su importancia no es grande a la hora de realizar una visita, visualmente ofrece un acabado más completo, y que dota de mayor realismo al edificio. En la *Ilustración 77* se puede contemplar el diseño de exterior de la sala de mantenimiento de la maquinaria de los ascensores, así como su situación en el edificio.

Otro aspecto importante a tener en cuenta ha sido la realización de los techos de los talleres, fácilmente reconocibles por su color amarillo. Este techado especial, compuesto por múltiples primas triangulares, se encuentra localizado sobre los talleres de la parte trasera (la parte más cercana al edificio 7, Juan Benet), y en el lateral del edificio colindante con la Calle Butarque (y situado sobre los laboratorios de Biomedicina).

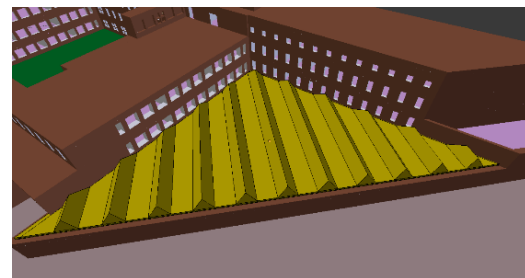
Para la realización de esta parte, se comprobó mediante las imágenes de Google Maps el número de prismas que componían cada techo y su apariencia, de tal manera que el techo resultante no otorgase ninguna discrepancia con la realidad (ya que los planos poseían una estructura con un número inferior de salientes, y el resultado obtenido daba la sensación de estar inacabado). En la *Ilustración 78* se ve detalladamente la estructura del techo situado sobre los laboratorios de Biomedicina y el trabajo realizado para dotarlo de la apariencia real.

Para completar la apariencia frontal del edificio, y que la sensación de inmersión del visitante sea completa, faltaría colocar las columnas exteriores de la fachada y las columnas que forman el soportal. Estas columnas se encuentran en la zona común de paso, por lo que son un elemento reconocible para cualquier persona que haya estado en el campus e imprescindibles si se quiere un modelo realista. En la *Ilustración 79* se ve con claridad como las columnas forman el límite del soportal por un lado, y la pared por el otro.



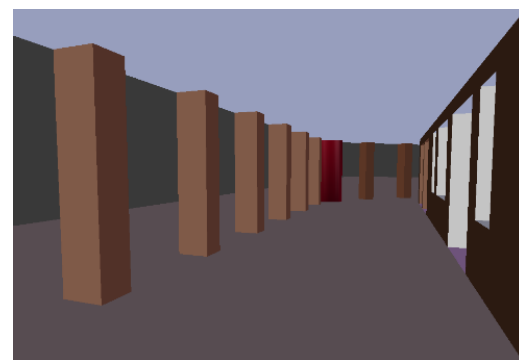
*Ilustración 77: Detalle de la sala de maquinaria del ascensor*

Detalle de la construcción de la sala de maquinaria de los ascensores situados en la azotea.



*Ilustración 78: Detalle de la estructura del techo de los talleres*

Se observa el modelado de los techos amarillos característicos de los talleres.



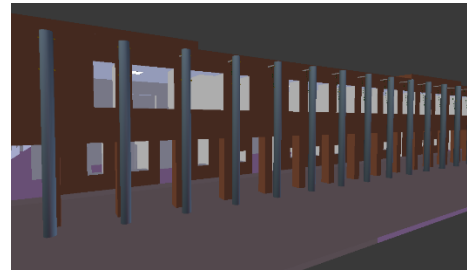
*Ilustración 79: Detalle de las columnas del soportal*  
Vista del soportal situado en el frente del edificio Betancourt, donde se observan tanto las columnas a un lado como la pared al opuesto.



Primero se han creado todas las columnas de ladrillo del soportal y la columna redonda situada en la salida del “pasillo E” siguiendo las medidas que nos ofrece el plano de la planta baja, de tal manera que el soportal crea un “pasillo” desde el que se accede tanto al edificio como a los establecimientos exteriores (reprografía, banco, agencia de viajes y espacio de estudiantes).

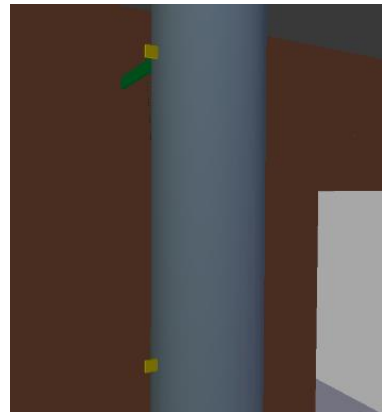
Para terminar con la apariencia frontal, se han modelado las columnas redondas que se encuentran delante de la fachada del edificio. Estas columnas se encuentran situadas justo delante de los pilares cuadrados que se acaban de crear, con una correspondencia exacta de 1:1 (por cada columna cuadrada de ladrillos existe una columna redonda). Estos nuevos componentes llegan hasta el final de la primera planta, y además poseen un enrejado que proporciona sombra a las salas que se encuentran en este lateral del edificio. Para el trabajo realizado en Blender, se han creado todas las columnas redondas que bordean y dan el aspecto final al edificio (*Ilustración 80*), junto con las sujeciones donde irá enganchado el enrejado (*Ilustración 81*), que será añadido posteriormente en el motor de juego.

Por último se ha creado el acceso desde el exterior al parking. Esta parte consta de la rampa asfaltada que comunica el nivel de calle con el sótano donde se encuentra el aparcamiento, y el césped que complementa la bajada. También se ha diseñado una puerta de acceso al parking, ya que tras realizar los “agujeros” para puertas y ventanas mencionados anteriormente, quedaba la pared con espacio disponible para la puerta automática de entrada al aparcamiento. La *Ilustración 82* muestra las tres partes mencionadas, la rampa en color gris, el césped el color verde, y la puerta automática en color rojo.



*Ilustración 80: Columnas de la fachada frontal*

Vista frontal del ed. Betancourt con las columnas redondas en su fachada.



*Ilustración 81: Detalle sujeciones del enrejado*

Vista en detalle de las sujeciones que unen el enrejado con las columnas redondas y la fachada.



*Ilustración 82: Rampa de acceso al aparcamiento subterráneo*

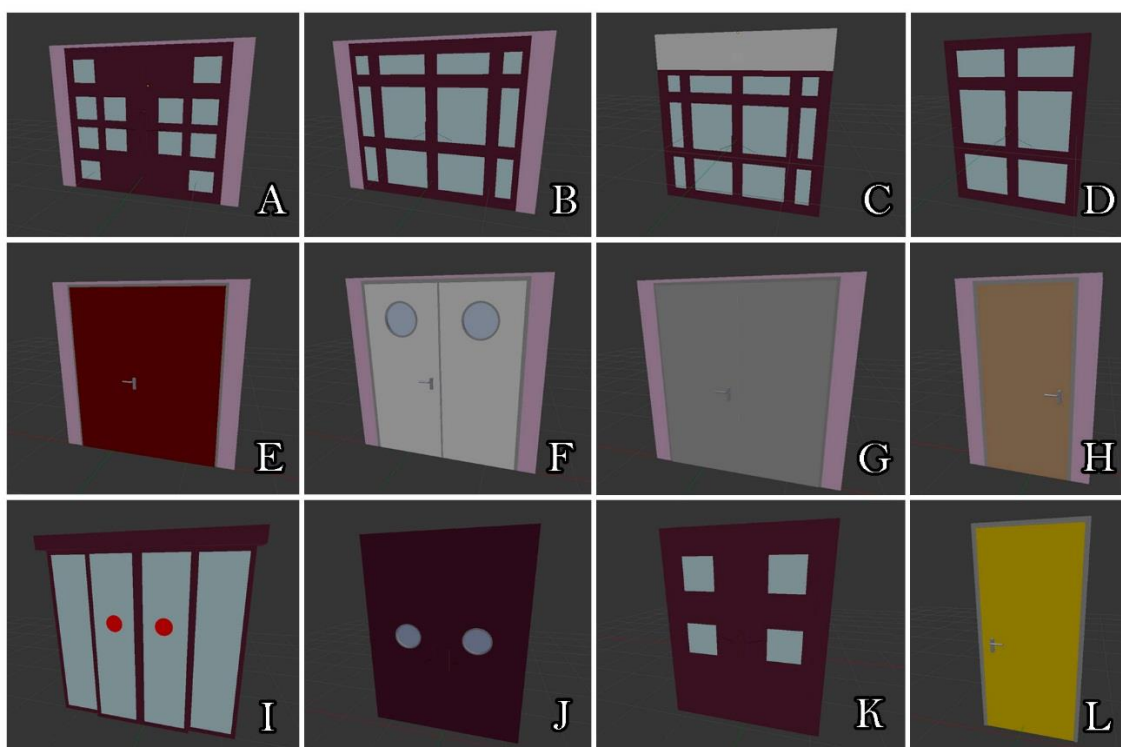
Detalle del acceso al aparcamiento, donde se observa la rampa, el césped y la puerta automática.

#### 4.1.2. Decoración interior

Tras haber realizado un modelado completo del edificio, tanto interna como externamente, debemos empezar a construir los distintos objetos que terminarán de dar el toque de realismo y cercanía a nuestra visita. En este punto, podríamos movernos libremente por cualquier parte, pero no existen puertas que delimiten las aulas o barandillas que nos impidan caernos (por poner algunos ejemplos), por lo que a partir de ahora se crearán los distintos elementos que poblarán nuestra visita.

El primer elemento en el que se ha puesto atención ha sido en las puertas. El edificio posee múltiples modelos diferentes de puertas. Entre ellas se pueden encontrar las puertas de acceso a las aulas, a talleres, a laboratorios, baños, de acceso a las escaleras, a los patios... Cada una cuenta con sus características propias (alto, ancho, molduras, refuerzo metálico, cristales, material en el que está construido...), por lo que se mostrarán algunos modelos para que se pueda ver el trabajo realizado.

En la *Ilustración 83* se pueden ver muestras de varios modelos de puertas realizados, los cuales serán explicados después de la imagen para poder reconocer los diseños fácilmente.

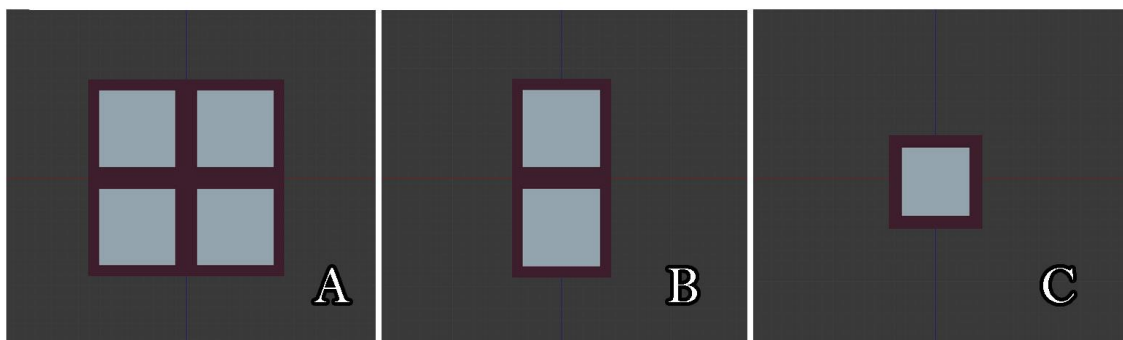


*Ilustración 83: Distintos tipos de puertas modelados*

- **A:** Puerta de acceso a los patios centrales que comunican con los pasillos.
- **B:** Puerta de acceso al patio triangular que da acceso a las aulas 1.0.H02 y H03 desde el “paseillo E”.
- **C:** Puerta de paso entre los pasillos y el hall existente nada más pasar la puerta de acceso de la calle.
- **D:** Puerta de acceso desde la calle al interior del edificio.
- **E:** Puerta de acceso a los talleres y a algunas salas del pasillo principal (compuesto por los “pasillos A-B-C-D”).
- **F:** Puerta que separa las escaleras situadas en los extremos del edificio del pasillo principal.
- **G:** Puerta de acceso a las aulas teóricas, salas de ordenadores y laboratorios.
- **H:** Puerta de baños y vestuarios.
- **I:** Puertas automáticas.
- **J:** Puerta cortafuegos situada en el pasillo principal entre las “pasillo C” y “pasillo D”.
- **K:** Puerta de acceso desde la calle a los distintos establecimientos (Agencia de viajes, banco, reprografía, etc.) y a las aulas de la “pasillo H”.
- **L:** Puertas de acceso a las salas situadas en la primera planta en la zona de los talleres.

Otro de los objetos realizados han sido las ventanas. En la construcción del edificio se pueden detectar tres tipos diferentes de ventanas, con los que se realizan cuatro composiciones distintas, que se encuentran repartidas por las diferentes fachadas exteriores.

En la *Ilustración 84* se observa los tres tipos distintos de ventanas, construidas con la misma proporción para poder comparar el tamaño de cada una con el resto.

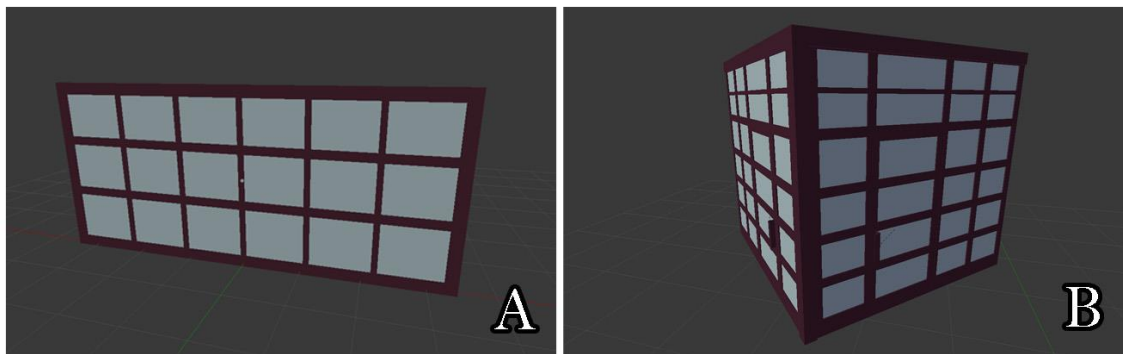


*Ilustración 84: Tipos de ventanas*

- **A:** Ventana grande situada principalmente en las aulas, ideal para permitir el acceso de gran cantidad luz.
- **B:** Ventana con la misma altura que la ventana anterior, pero más estrecha. Se encuentra principalmente en despachos.
- **C:** Ventana con la mitad de altura y mismo ancho que el modelo anterior, y con un cuarto de tamaño que la primera ventana mostrada. Se encuentra principalmente en los pasillos.

Siguiendo con la construcción de ventanas, y debido a su estructura similar, pasamos a la creación de las cristaleras. Existen dos tipos de cristaleras características en el edificio. El primer tipo de cristaleras se encuentran situadas en la intersección de los “pasillos C-D-G”, y aportan visibilidad desde el interior del pasillo tanto a uno de los patios internos, como al rocódromo y la entrada que une el edificio Betancourt con el edificio Benet. El segundo tipo de cristaleras se encuentran en el patio triangular formado por los “pasillos A-E-H”, y separa la puerta de acceso a las aulas 1.0.H02 y 03 con el patio.

En la *Ilustración 85* se pueden ver los dos tipos de cristaleras, a la izquierda la que nos podemos encontrar situada desde la primera a la tercera planta a ambos lados del final del “pasillo C” (justo en la intersección de los “pasillos C-D-G”), y a la derecha la cristalera de que separa el patio con la entrada de las aulas.



*Ilustración 85: Modelado de las cristaleras*

Cambiando el tipo de elemento a construir, se ha pasado a realizar el modelado de los ascensores. Se puede observar que en los ascensores se han cuidado todos los detalles, como se puede apreciar en la *Ilustración 86*. Se distinguen las dos puertas que componen cada ascensor, junto con la botonera para llamarlos, las pantallas indicadoras de la planta y los bordes de distinto material que separan cada ascensor del otro.



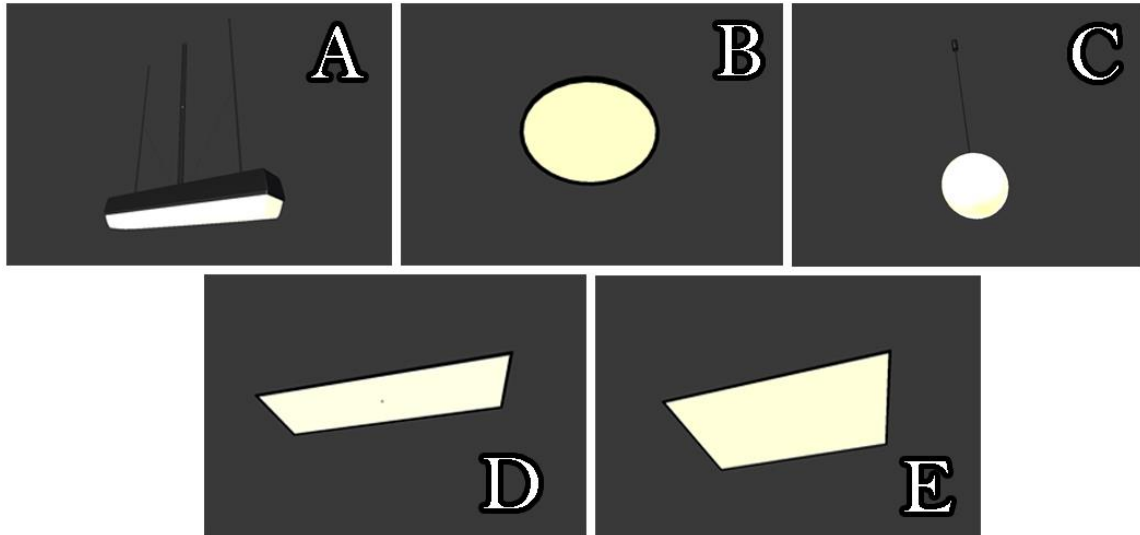
*Ilustración 86: Modelado de los ascensores*

Las barandillas es otro de los objetos a tener en cuenta. Su localización los sitúa en el pasillo central de la planta primera, segunda y tercera, junto a la pared del rocódromo, y en la primera planta en la pasarela existente en los talleres, que sirve de unión entre el pasillo principal y las salas con apariencia exterior amarilla. En la *Ilustración 87* se puede ver en detalle la barandilla situada en una de las pasarelas de los talleres, que da acceso desde el edificio principal hasta las salas de trabajo.



*Ilustración 87: Barandillas situadas en la pasarela de los talleres*

Para terminar con los elementos principales, se mostrarán las luces repartidas por todo el edificio. Se han dividido en cinco tipos distintos, y cada tipo corresponde a una función concreta. En la *Ilustración 88* se pueden observar todos los tipos de iluminación que se ha utilizado en todo el edificio.

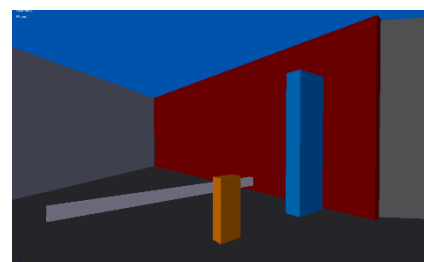


*Ilustración 88: Tipos de luces*

- **A:** Fluorescentes situados en el aparcamiento subterráneo. Compuestos por tres sujeciones al techo, la cubierta de plástico y la cubierta semitransparente.
- **B:** Plafón situado en los techos de los rellanos de las escaleras. La iluminación que aportan es menor, pero suficiente para los espacios pequeños en los que se pueden encontrar.
- **C:** Lámpara de techo situada a lo largo de toda la escalera principal.
- **D:** Iluminación compuesta por dos fluorescentes, situados a lo largo de todos los pasillos.
- **E:** Iluminación compuesta por cuatro fluorescentes (el doble del tamaño que el objeto anterior) y situadas en las aulas y laboratorios, al ser espacios con mayor distancia entre las paredes.

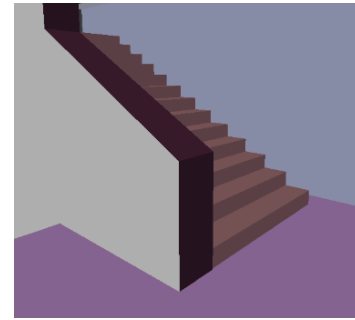
A partir de este punto, los objetos diseñados poseen un matiz más secundario, decorativo o unitario (sólo se observan en un punto en concreto del modelado), pero estos detalles permiten dotar de mayor realismo al conjunto del modelado por donde el visitante podrá dirigir su paseo virtual.

En el sótano, como se comentó al hablar del diseño de la entrada al parking, se ha creado una puerta automática para el acceso al aparcamiento. Además, también se han modelado las barreras de control que serán situadas nada más cruzar la puerta. En la *Ilustración 89* se distingue en rojo el modelado de la puerta de acceso, y en naranja y blanco la barrera de control de acceso al aparcamiento.



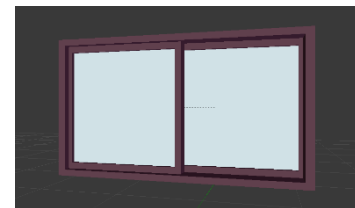
*Ilustración 89: Puerta del garaje y barrera de paso*

También se ha terminado de modelar la cubierta del pasamanos interior de las escaleras. En el diseño inicial se había creado toda la pieza del mismo color, por lo que se le ha añadido el pasamano metálico de color granate. Es un detalle mínimo, pero que aporta mucho contraste. En la *Ilustración 90* se destaca el pasamano de color granate que recorre todo el borde de la barandilla central de las escaleras.



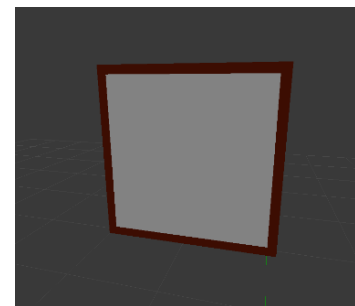
*Ilustración 90: Cubierta metálica del pasamano*

Otro elemento diseñado ha sido la ventana de conserjería, a través de la que los alumnos y los conserjes interactúan. Esta ventana es de apertura lateral, por lo que su diseño es totalmente distinto a las ventanas creadas anteriormente. En la *Ilustración 91* se pueden observar las dos ventanas de apertura lateral que componen este objeto.



*Ilustración 91: Ventana de conserjería*

Por último, y para poder colocar los carteles identificativos por todo el edificio, se han creado los soportes en los que posteriormente irá cada letrero correspondiente (carteles de los baños, de las plantas, los mapas...). Todos los soportes se componen de un pequeño marco de color rojo, en el que dentro irá situada la textura correspondiente a cada tipo de cartel (*Ilustración 92*). También será necesario cambiar el tamaño dependiendo del tipo de información que se quiera indicar, ya que no es el mismo espacio el requerido para un indicativo de planta que del mapa de situación del visitante.



*Ilustración 92: Soporte de carteles*

#### 4.1.3. Texturizado de modelos

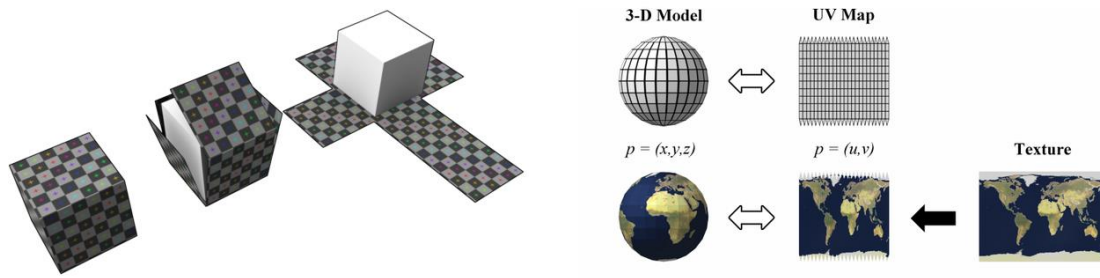
En esta sección se explicará el proceso seguido para dotar de realismo los modelos creados, así como los diferentes materiales y texturas necesarios en cada objeto. Todas las texturas utilizadas en este proyecto han sido extraídas de distintos repositorios gratuitos<sup>1</sup>. Para poder entender el trabajo realizado es importante conocer la técnica de texturizado UV Mapping, imprescindible para entender cómo se ha dotado de realismo a los objetos mediante texturas.

La técnica de texturizado UV Mapping permite establecer una correspondencia entre los vértices de un objeto y las coordenadas de una textura. Mediante esta técnica no es necesario crear un fichero o imagen para representar cada una de las texturas que se aplicará en las distintas

<sup>1</sup> Repositorios de texturas: [cgtextures.com](http://cgtextures.com), [texturemate.com](http://texturemate.com), [lemog.fr](http://lemog.fr), [mayang.com](http://mayang.com).



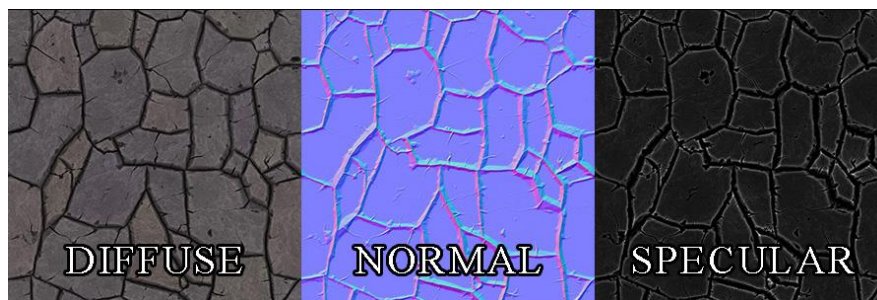
superficies que forman un objeto. En otras palabras, en un mismo fichero se puede representar la "piel" que cubrirá diferentes regiones de una superficie tridimensional (*Ilustración 93*). [22]



*Ilustración 93: Representación gráfica de la técnica UV Mapping*

Además, hay que saber que una texturización completa no se compone simplemente de una imagen bidimensional que recubra la figura, sino de tres imágenes con distintas características que componen la textura final (*Ilustración 94*). Estas tres características son:

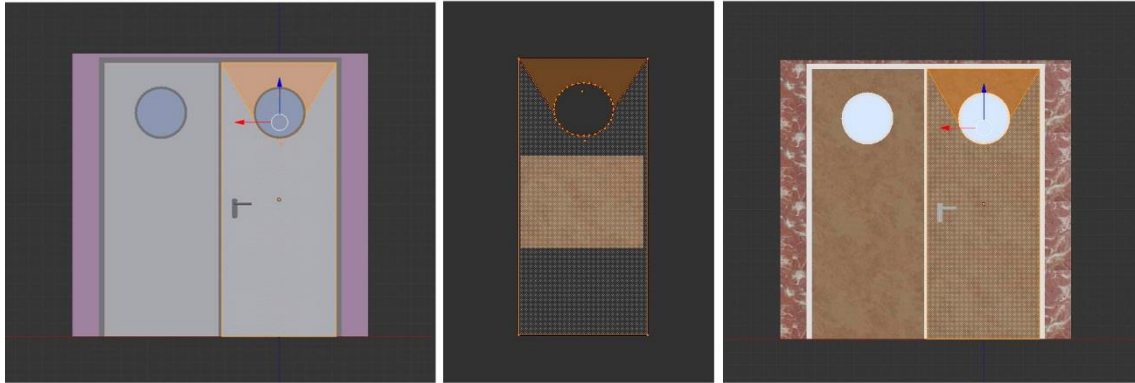
- **Diffuse:** Es la imagen original que queremos usar como textura del objeto. La información que aporta es básica, como las formas y los colores.
- **Normal:** Partiendo de la imagen original (diffuse), esta textura contiene la información referente al relieve y luminosidad de cada parte de la imagen, y además añade la información referente al cálculo de las normales. Se emplea, entre otras funciones, para simular arrugas, poros, o superficies rugosas. La imagen resultante es característica por sus tonalidades moradas y azules.
- **Specular:** Partiendo de la imagen original (diffuse), esta textura contiene la información referente al cálculo de la reflexión de la luz. Se utiliza para calcular qué partes del objeto deben ser más opacos (por ejemplo tierra) y en cuales el material refleja más la luz (como pueden ser metales o cristal). La imagen resultante es característica por ser una imagen formada por blanco, negro, y escala de grises.



*Ilustración 94: Mapas de texturas*

A pesar de que la texturización también se puede realizar con el motor de juego, se ha optado por texturizar los modelos en Blender debido a la mayor versatilidad que ofrece el programa de modelado para configurar a la perfección cada polígono, mientras que Unity 3D sólo permite ajustar parámetros generales (como la escala de la textura en el eje X y eje Y) en toda la figura.

Teniendo claro estos conceptos, y usando el método Smart UV Project de Blender, podremos adaptar cualquier textura disponible al objeto deseado. Esta opción nos crea distintos polígonos que deberemos ajustar a las diferentes secciones de la textura seleccionada para obtener un resultado satisfactorio.



*Ilustración 95: Ejemplo de UV Mapping aplicado al diseño de una puerta*

En la figura *Ilustración 95* se puede apreciar como las caras seleccionadas (imagen de la izquierda) han sido colocadas sobre la textura elegida con el escalado que se ha considerado adecuado (imagen central), para que ésta se distribuya sobre la superficie, observando el resultado final en la imagen de la derecha.

Blender ofrece la posibilidad de que las texturas con las que se trabaja se repitan en modo “mosaico”, por lo que teniendo una pequeña textura se pueden reproducir grandes superficies, como paredes o suelos. Gracias a esta opción, se reduce considerablemente la memoria utilizada en el proyecto, ya que no es necesario tener una textura tan detallada como para aplicarla en superficies grandes, sino que se repetirá las veces necesarias hasta que la textura quede cubierta.

Para poder representar bien todas las texturas y debido a que se aplican a componentes de gran tamaño, se han dividido los objetos que constituyen cada elemento del modelado según el material que lo forma. Esto es debido a que los materiales se componen de una imagen que se repite en mosaico, por lo que para configurar correctamente la correspondencia de todos los vértices, ha resultado más sencillo utilizar esta técnica.

También se han configurado las características de los “mapas de normales” (profundidad y rugosidad) y de los mapas especulares (intensidad de la reflectividad de la luz), ya que aunque en el motor gráfico es necesario volver a indicar la ruta de los mapas de texturas, es necesario dejar configuradas estas características para que se apliquen bien las propiedades posteriormente.

De igual manera, en el motor gráfico se puede modificar la tonalidad de algunos materiales, por lo que no todas las texturas aplicadas en Blender tienen que poseer la apariencia que finalmente adoptaran durante la visita virtual. Esta característica será explicada con mayor detalle (hay que explicarla bien) más adelante en la sección “4.2. Motor de juego”.

A pesar de todo lo explicado anteriormente, hay veces en las que no es posible gestionar el texturizado de un objeto directamente desde la imagen obtenida, y esa imagen debe ser

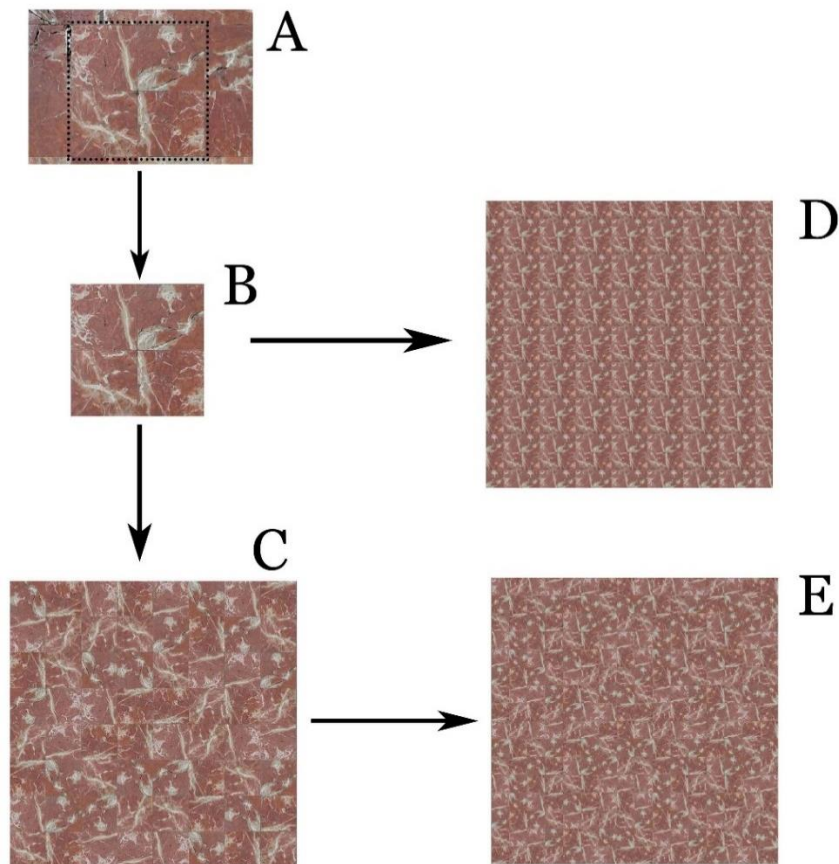
retocada mediante un editor de imágenes. En este proyecto el programa elegido ha sido Adobe Photoshop (como se ha explicado en el subapartado “Otros programas” de la sección “2.3. Diseño 3D”).

En la mayoría de los casos, la edición de las texturas se ha reducido a ligeros retoques, como eliminar partes sobrantes de una textura o tapar defectos y roturas del material original. En la *Ilustración 96* se puede ver cómo partiendo de la imagen una barrera de aviso se ha construido una textura “limpia”, en la que se han arreglado los desperfectos y se ha ajustado para poder utilizarla como mosaico horizontal.



*Ilustración 96: Creación de la textura "Barrera"*

Pero en otras ocasiones también se ha necesitado trabajar más concienzudamente en las texturas donde la labor de edición ha sido grande y costosa, como es el caso de los suelos de mármol. El recurso inicial que se tenía era una imagen parcial de un suelo de mármol, y a partir de esa imagen se ha creado una textura elaborada en la que se arreglan desperfectos y se genera un gran mosaico en el que se evitan repeticiones. A continuación se explica el proceso realizado para generar la nueva textura, que podrá ser seguido visualmente con la ayuda de la *Ilustración 97*.

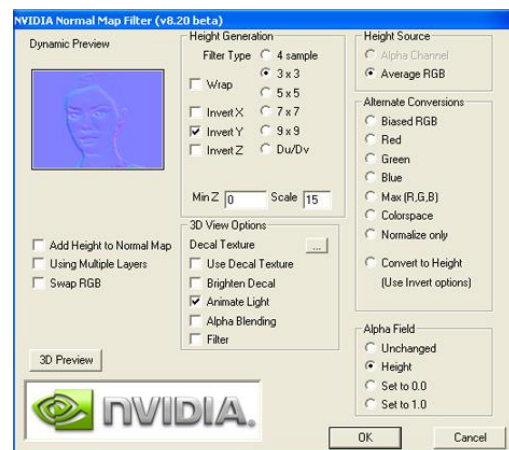


*Ilustración 97: Proceso de creación del suelo de mármol*

- **A - Imagen inicial:** El recurso inicial del que se disponía. Es una imagen de un suelo de mármol en la que se ven cuatro baldosas completas y varias losas cortadas en los bordes, lo que impide su utilización como mosaico al no coincidir los extremos. Además, las baldosas de mármol tienen desperfectos.
- **B – Textura 2x2:** Se han eliminado las baldosas incompletas de los bordes, dejando únicamente las cuatro centrales y manteniendo la junta entre cada losa. Además, se han arreglado varios desperfectos y manchas de suciedad presentes en la imagen inicial.
- **C – Textura 8x8:** Usando como base la “textura 2x2” anterior, se ha dividido la imagen en las 4 baldosas independientes, que han sido repartidas aleatoriamente por toda la textura. Para evitar repetir patrones, cada losa se ha colocado rotándola sobre sí misma en cada iteración, y se ha tenido en cuenta el que no se duplicasen nunca las baldosas que se encuentran a su alrededor, formando una textura de 8x8 losas totalmente nueva.
- **D – Mosaico realizado con la textura 2x2:** Mosaico resultante de utilizar en una superficie grande la textura 2x2. Como se puede observar, el patrón resultante es muy repetitivo y ofrece poco realismo.
- **E – Mosaico realizado con la textura 8x8:** Mosaico resultante de utilizar en una superficie grande la textura 8x8. Con esta textura se genera un patrón realista que no aporta sensación de repetición.

Para completar la técnica de texturizado mediante imágenes simples o “diffuse”, se han tenido que crear texturas que precisen relieve para aquellos casos en los que sea necesario su correspondiente mapa de normales (normal map) y su mapa especular (specular).

Para la generación de los mapas de normales ha sido necesario utilizar la herramienta para desarrolladores “Nvidia Texture Tools for Adobe Photoshop”<sup>2</sup>. Como indica su propio nombre, sólo es compatible con equipos que utilicen Photoshop y trabajen con tarjetas gráficas Nvidia. Se instala como un plug-in y trabajando de la misma manera que si se aplicase un filtro, permite crear el mapa de normales de una imagen que luego se utilizará en Blender para dotar de relieve a la textura, pudiendo modificar todas las características necesarias (Ilustración 98).



*Ilustración 98: Generación del mapa de normales con la herramienta de Nvidia*

Por su parte, para generar los mapas especulares también ha sido necesario modificar el recurso original. Las imágenes con características especulares se generan partiendo de la imagen diffuse, siendo necesario eliminar la información de color que contiene, dejando la textura solamente con una gama de colores en blanco y negro o escala de grises. De esta manera, Blender es capaz

<sup>2</sup> Disponible en: <https://developer.nvidia.com/nvidia-texture-tools-adobe-photoshop>

de reconocer qué partes reflejan más o menos la luz según su proximidad en la tonalidad al blanco (material muy reflectante) o al negro (opaco).

Aunque hasta este momento se ha hablado de la texturización como técnica para aplicar realismo a los objetos, no todos los modelos deben poseer necesariamente una textura. En el caso de los cristales, se ha utilizado un material azul transparente que refleje la luz. De esta manera, se reduce la carga de procesamiento de una imagen, reduciendo el tiempo de ejecución del material al ser representado sobre el objeto.

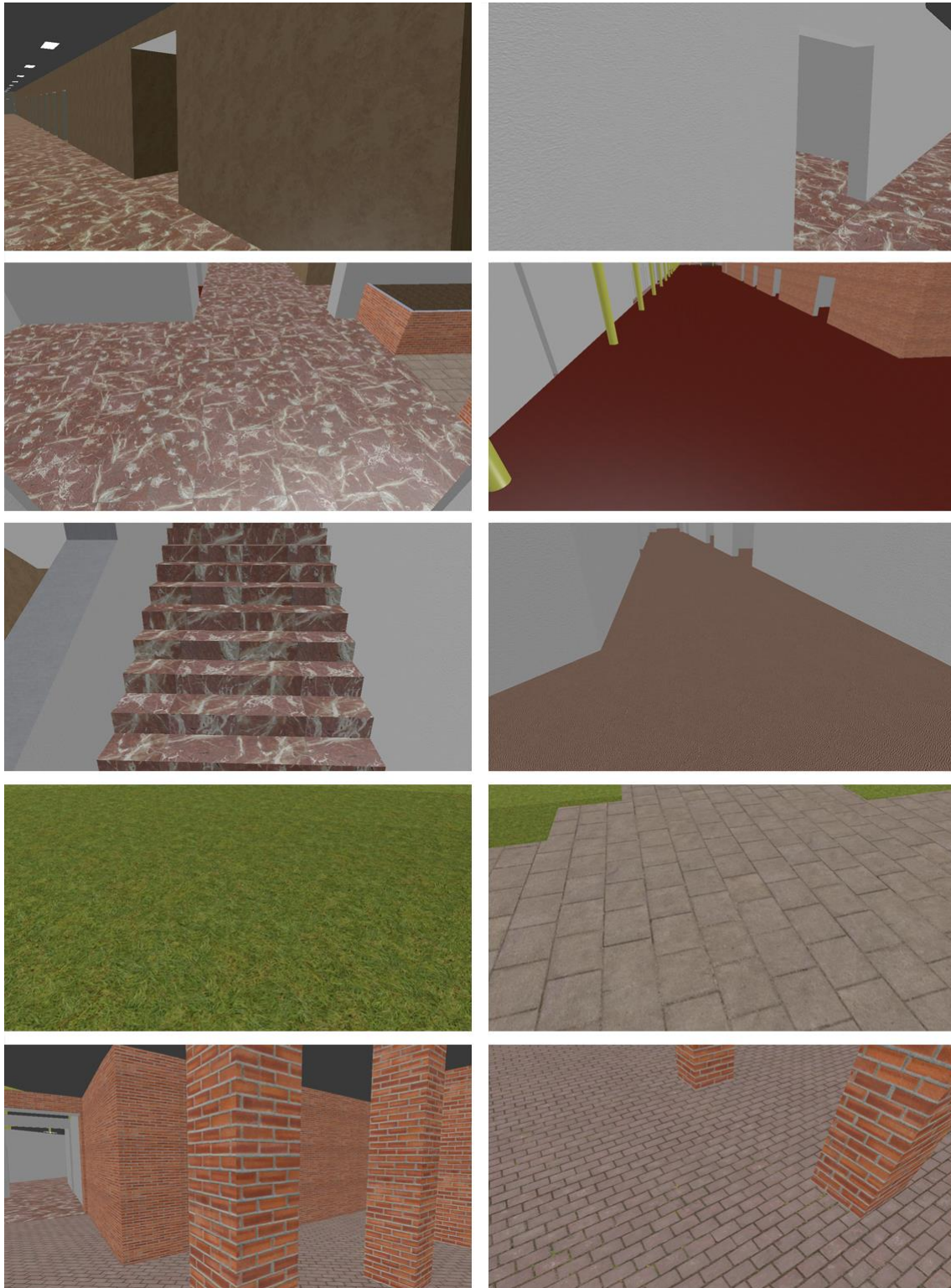
Tras haber explicado cómo se proporciona realismo a los modelados a través de la texturización, es el momento de mostrar la apariencia definitiva de los elementos creados, de tal manera que se pueda apreciar el trabajo realizado. Para ello, se muestran desde la *Ilustración 99* hasta la *Ilustración 101* las diferentes texturas y materiales utilizados.



*Ilustración 99: Muestra de diferentes tipos de texturas para los carteles del edificio*

Los carteles distribuidos por el edificio corresponden a indicaciones de la planta en la que se está, los baños, indicativos y de autorización de paso, los mapas de situación y el referente al nombre de la escuela.





*Ilustración 100: Muestras de diferentes tipos de texturas*

Los nombres de las texturas son: "Pared marrón", "Gotelé paredes", "Mármol suelo", "Suelo talleres", "Escaleras", "Suelo biomedicina", "Césped", "Suelo patios", "Ladrillos" y "Suelo calle".





*Ilustración 101: Muestras de diferentes tipos de texturas*

Los nombres de las texturas son: "Suelo metálico", "Pintura plástica", "Pintura metal amarillo", "Columnas hormigón", "Suelo asfalto", "Barrera" y "Soporte barrera", "Puerta parking", "Pared gris aparcamiento" y "Pared roja aparcamiento", "Azotea" y Material "Cristal".

## 4.2. Motor de juego

La sección referente al motor de juego se ha realizado con Unity 3D, como se justificó en capítulos anteriores. La versión comercial disponible al iniciar el proyecto era la v4.6, por lo que todo el trabajo se ha realizado con las opciones disponibles en la versión gratuita que ofrecía Unity 3D en esa versión. También existe una versión Pro, sin ninguna restricción en las opciones a usar, pero siendo necesario comprar una licencia tanto para el uso de estas características, como para la comercialización del producto.

El motor de juego unifica todos los diseños realizados con Blender para poder construir nuestro escenario para la visita, y además permite añadir el sistema de interacción usuario-aplicación. Además, también es el encargado de renderizar todo el escenario, gestionando el sistema de iluminación, el motor de físicas y colisiones, o los menús.

### 4.2.1. Importación de modelos

Los diseños importados en Unity 3D provienen de los generados con Blender anteriormente, los cuales se han mantenido en formato .blend (nativo de Blender), ya que como se ha explicado anteriormente, Unity 3D es capaz de detectar los cambios realizados en el modelo original y re-importar automáticamente el objeto, para tener siempre la versión más actualizada en el motor de juego.

El único posible inconveniente de trabajar con archivos .blend es que es necesario tener una copia licenciada de Blender instalada en la máquina en la que se vaya a usar Unity. En nuestro caso, al utilizar el mismo equipo no supone ningún problema, pero es algo que hay que tener en cuenta si en el futuro se desea modificar el proyecto en otro ordenador diferente. [23]

### *Gestión de modelos*

En este punto inicial, nos encontramos con la ventana de la “escena” de Unity 3D vacía, por lo que añadimos y colocamos todos los diseños creados en Blender al proyecto. Al realizar todos los modelos en la misma escala y manteniendo las mismas dimensiones, no tendremos ningún problema con el escalado en Unity 3D, ya que la información de las medidas que componen cada objeto se guarda en el archivo .blend original.

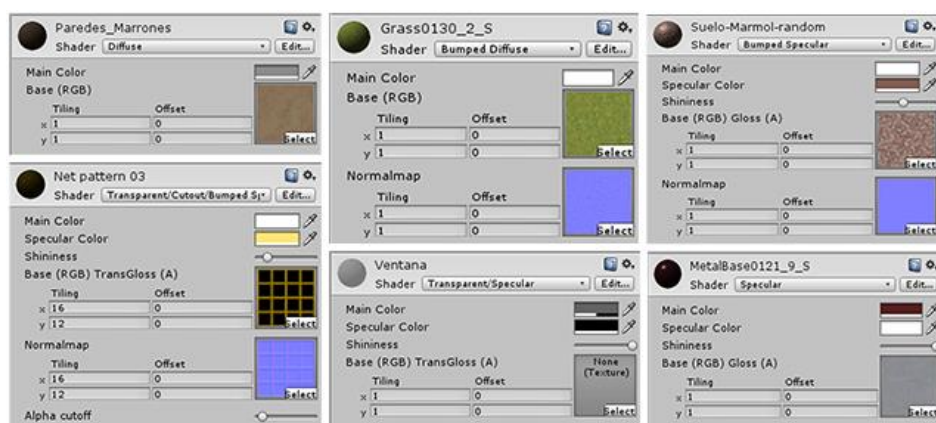
Se ha considerado oportuno dividir el modelado del edificio Betancourt por plantas, ya que al ser un archivo grande, si se realizaba alguna modificación el tiempo generado en re-importar el modelo en Unity 3D era bastante elevado.

### *Texturizado*

Tras importar los modelos en Unity 3D, es necesario volver a definir la ruta de las texturas para generar los materiales que compondrán cada objeto. Aunque Blender conserve la información referente al UV Mapping de la textura (polígonos en los que se ha aplicado, posición, escalado...), Unity 3D no es capaz de cargar la ruta, por lo que otorga por defecto a todas las texturas la propiedad “Diffuse”, y hay que volver a indicar la imagen que se convertirá en textura. También

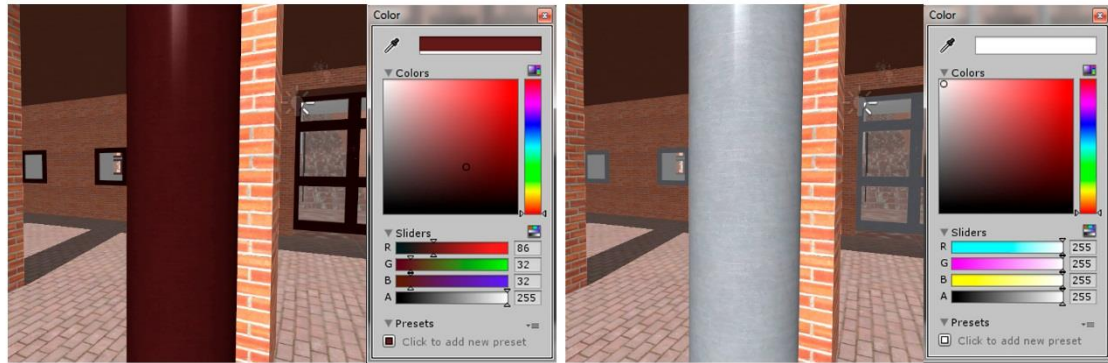
es necesario definir los shaders (conocidos como “sombreadores”, que son pequeños scripts que permiten configurar la forma en la que el hardware gráfico es ajustado para renderizar los materiales) que definirán cómo el motor tratará gráficamente cada textura. Para la configuración de las texturas se han utilizado los siguientes shaders:

- **Diffuse:** Shader básico por defecto. Es la imagen original, sin efectos.
- **Bumped Diffuse:** Compuesta por dos imágenes. La primera corresponde a la textura básica (diffuse). La segunda permite dotar de relieve a la textura y corresponde al mapa de normales.
- **Bumped Specular:** Además de la información solicitada en el shader “Bumped Diffuse”, hay que indicar el color de la especularidad (la tonalidad en la que se verá el reflejo) y el nivel de definición para el reflejo (si se quiere obtener una imagen en el reflejo nítida o borrosa) configurada mediante el parámetro Shininess.
- **Specular:** Hay que configurar los mismos parámetros que en Bumped Specular, pero sin indicar el mapa de normales, ya que se considera que esta textura es lisa.
- **Transparent/Specular:** Shader utilizado para los cristales, convierte al objeto en un material transparente y con reflectividad. La información necesaria para configurar este shader es la misma que la configuración Specular, con la particularidad de que se puede dejar sin indicar una textura básica para que el objeto sea más parecido a un cristal pulido.
- **Transparent/Cutout/Bumped Specular:** Este shader se utiliza para las rejillas, en las que la textura tiene transparencias, de tal manera que haya partes que se vean y partes que no (propiedad Cutout). Los elementos necesarios para configurar este shader son los mismos que con Bumped Specular, con el requisito de que la textura debe estar guardada en formato .PNG (este formato admite guardar imágenes con partes transparentes, mientras que otros formatos como .JPG rellenan los huecos vacíos con un color predeterminado).



*Ilustración 102: Propiedades de los distintos tipos de "shaders"*

En la *Ilustración 102* se pueden observar las características de todos los tipos de shaders comentados anteriormente. Además, existe otra propiedad llamada “Main Color”, en la que se puede definir un color básico que “teñirá” la textura con ese color. Si se deja en blanco, la textura mantiene sus colores intactos. En la *Ilustración 103* se puede ver a la izquierda cómo la propiedad Main Color está configurada para teñir la columna de rojo mientras a la derecha está “en blanco”, pudiéndose observar la textura en su color original.



*Ilustración 103: Comparativa de la propiedad "Main Color"*

#### 4.2.2. Uso de recursos externos

Además de los modelos creados en Blender, también se han utilizado recursos gratuitos disponibles en la red, tanto en la tienda Asset Store de Unity 3D como en repositorios gratuitos de diseños tridimensionales<sup>3</sup>.

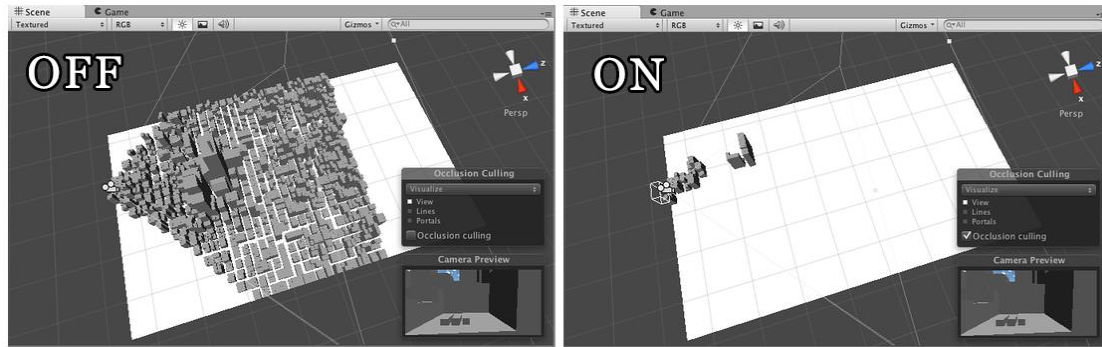
La finalidad de los recursos externos utilizados es la de crear una ambientación realista, pero teniendo en cuenta que son un componente secundario. Por este motivo, el nivel de detalle de algunos de estos objetos es inferior al resto de elementos creados, ya se le ha dado mayor importancia a un renderizado rápido que a incrementar la calidad visual general que se alcanza al utilizar estos modelos.

De la primera selección realizada, se han descartado todos los modelos que ralentizaban el renderizado y no permitían un movimiento fluido. Estos modelos eran diseños con mucho detalle, superando en algunos casos los 65.536 polígonos (Unity no es capaz de gestionar un objeto individual con tanta definición).

Al trabajar con la versión gratuita, es muy importante no sobrecargar el escenario con diferentes recursos y objetos, ya que no se pueden utilizar características que optimicen el renderizado, como la técnica “occlusion culling”, al estar disponible únicamente en la versión Pro. Si se configura correctamente esta opción, permite al motor de juego deshabilitar el renderizado de objetos cuando no se muestran en pantalla porque se encuentran tapados por otros elementos, mientras que con esta característica deshabilitada el motor renderiza todos los modelos y recursos posicionados en el escenario, independientemente de si son visibles o no.

<sup>3</sup> Repositorios de modelos 3D: [tf3m.com](http://tf3m.com), [archive3d.net](http://archive3d.net), [3delicious.net](http://3delicious.net).





*Ilustración 104: Comparativa de renderizado con y sin "occlusion culling"*

En la *Ilustración 104* se puede observar la diferencia de renderizado cuando se trabaja con y sin "occlusion culling", y así entender por qué es importante situar pocos objetos cuando no se dispone de la versión Pro, ya que la experiencia sería poco fluida y con saltos al no poder gestionar todos los elementos disponibles.

### *Asset Store de Unity*

La fuente principal de recursos ha sido la Asset Store de Unity, ya que los recursos están optimizados para ofrecer el mejor acabado posible afectando al mínimo el rendimiento. A pesar de poseer infinidad de recursos en la tienda oficial, se ha optado por trabajar sólo con los paquetes gratuitos. Los recursos disponibles no se reducen a modelos, sino que también existen materiales, scripts, animaciones, herramientas para audio... y cualquier elemento que pueda ser utilizado con Unity, existe en la Asset Store.

A continuación se indicarán los recursos utilizados, así como la finalidad que tienen y por qué se han empleado en la visita. En esta sección sólo se hablará de los recursos "visuales", como son materiales y modelos, ya que los recursos de código (como son los scripts) serán explicados más adelante.

### *Materiales*

Se ha utilizado un material especial perteneciente al paquete "Yughues Free Grids & Nets Materials" para crear las rejillas características de la fachada frontal del edificio. Como se ha explicado anteriormente en la sección "Texturizado", se utiliza un tipo de shader que permite transparencias, por lo que nuestras rejillas permiten ver el objeto que hay detrás, pero a su vez se muestra la figura y la sombra que se proyecta.

### *Modelos*

Se han utilizado múltiples recursos, tanto integrados en Unity 3D por defecto como pertenecientes a la Asset Store, de esta manera se podrá recrear la ambientación existente en el edificio de una manera más fiel.

*Tree Creator:* Es el único de los recursos utilizados que viene integrado por defecto en el paquete "Standard Assets" de Unity 3D. Este asset permite la creación de árboles con distintas características para simular bosques y evitar modelos repetidos. En este caso, se ha utilizado el

modelo básico para simular los árboles situados en los diferentes patios internos que componen la estructura del edificio Betancourt.

*Props for the Classroom:* Este paquete contiene distintos objetos ideales para la creación de clases (tanto aulas de teoría como aulas de ordenadores). Entre otros elementos incluye pizarras, mesas, sillas, ordenadores o proyectores.

*FPS Hangar:* (Gratuito en el momento de su descarga, de pago en la actualidad) Paquete de recursos compuesto por distintos elementos decorativos en la construcción de un hangar. Los modelos disponibles se utilizarán para componer los talleres, como son bidones, cajas, numeración para el suelo, taquillas, contenedores, cubos de basura, lámparas, paneles controladores de luz o palets de madera.

*Coches:* Se han utilizado varios modelos existentes en la Asset Store pertenecientes a Unity Technologies. Aunque estos modelos corresponden a coches clásicos, cumplen su funcionalidad en el aparcamiento de la misma manera que si fuesen coches modernos.

*Vending y Snack machine:* Se han usado varios modelos de máquinas dispensadoras de bebida y comida para situar a lo largo del pasillo central, tal y como sucede en la realidad.

*Japanese Rubbish Bin:* Papelera para los pasillos. Aunque no es un modelo parecido a la realidad, cumple las funciones de simular una papelera y que sea reconocible por el visitante.

*Low Detail Forklift:* Situada en los talleres, la carretilla elevadora ayuda a la composición del taller con un mayor número de elementos distintos.

### **Repositorios Online**

Como hay modelos que no se encuentran gratuitamente en la Asset Store, se ha buscado en distintos repositorios online de modelos 3D. Las tres bases de datos consultadas han sido TF3M, Archive3D y 3Delicious. Como son modelos “externos” es necesaria una revisión de cada uno para volver a aplicar las texturas correctamente, pero como no se busca un acabado detallado sino una ayuda visual, se ha optado por aplicar una capa base de color sin profundizar en las texturas para evitar que produjesen fallos en el rendimiento.

*Elevador de coches:* El elevador de coches se ha utilizado para el taller de mecánica, ya que se realizan prácticas con coches.

*Password panel:* Panel numérico de control de acceso colocado en la puerta de acceso a biomedicina y al centro de cálculo situado en el “pasillo G”.

*Cajero automático:* Cajero automático situado en el pasillo al que se accede desde la puerta que comunica con el edificio Sabatini. Ha sido editado para añadir los logotipos y colores del modelo real y que así pueda ser reconocible.

*Coches:* Se descargaron más de una veintena de modelos de vehículos distintos, pero debido a fallos al cargar partes del modelado o a su elevado detalle, al final la selección de vehículos ha descendido hasta la docena de modelos. Como se ha comentado anteriormente, no se busca gran realismo en estos modelos, ya que son “añadidos” al proyecto y no partes fundamentales de la visita.

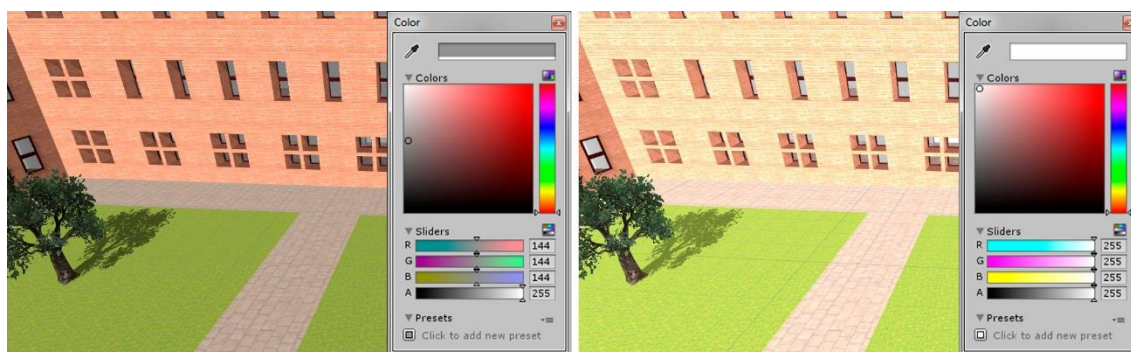


### 4.2.3. Iluminación

La iluminación es un pilar fundamental a la hora de crear cualquier escenario, y más en el caso de una visita virtual en la que una iluminación equivocada puede arruinar la experiencia. Para hablar esta sección se dividirá la explicación entre iluminación exterior e iluminación interior.

#### *Iluminación exterior*

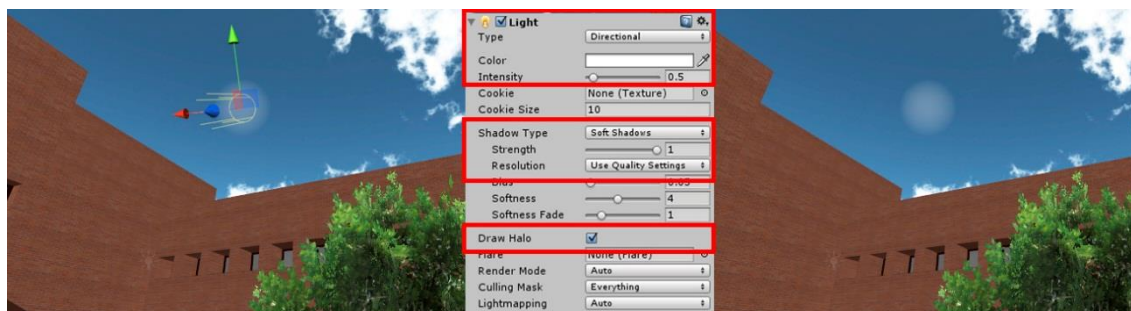
El primer paso ha sido configurar una iluminación mínima ambiente (no crea sombras) común a todo el escenario. Se ha configurado como una luz blanca tenue, que aunque consigue que exista una ligera visibilidad en todos los rincones, no llega a modificar la apariencia de los objetos por ser demasiado fuerte. En la *Ilustración 105* se aprecia el cambio existente entre la luz ambiente tenue escogida (a la izquierda) y una luz ambiente fuerte (a la derecha), comprobando que demasiada luz ofrece un acabado con saturación de color.



*Ilustración 105: Comparativa Luz Ambiente*

Después de configurar la iluminación ambiente, se va a crear iluminación exterior. En este caso, la visita virtual se va a realizar durante un día soleado, por lo que no es necesario crear luces de farolas, simplemente la iluminación proveniente del sol.

Para imitar al sol, se ha utilizado una luz direccionada (simula una luz colocada infinitamente lejos y que afecta a toda la escena). Como se pretende emular al sol, se ha configurado la posición de la luz para que la incidencia sobre el edificio corresponda con la de mediodía, creando también un halo alrededor del foco para que el visitante pueda situar sobre el cielo el sol. Se le ha proporcionado una tonalidad blanca con una intensidad media (para evitar una saturación de luminosidad), así como la propiedad de que produzca sombras sobre los objetos en los que incida. En la *Ilustración 106* se pueden observar con detenimiento las propiedades de la iluminación exterior.

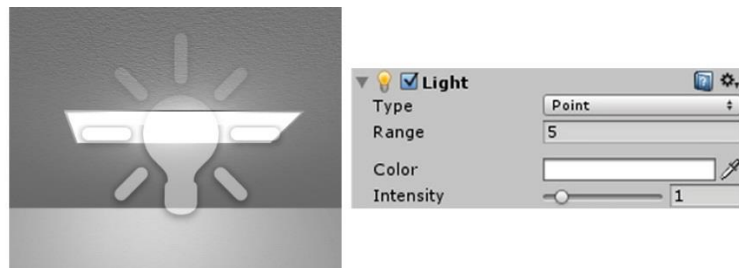


*Ilustración 106: Propiedades iluminación exterior*

### *Iluminación interior*

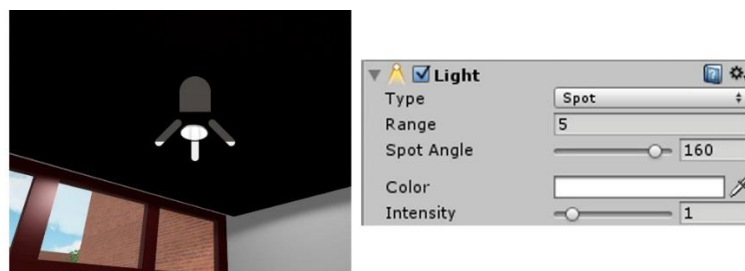
Para la iluminación interior del edificio se han utilizado diferentes tipos de iluminación correspondiente a las luces indicadas en la sección “4.1.2. Decoración interior”. El tipo de iluminación se ha dividido en iluminación mediante puntos de luz (conocido como point light, alumbra con la misma intensidad en todas las direcciones afectando a todos los objetos que se encuentren dentro de su rango), e iluminación mediante focos (conocido como spot light, alumbra en forma de cono todos los lugares que estén dentro de su proyección). El resto de opciones de iluminación como la proyección de sombras o la luz de área son opciones sólo disponibles en la versión de pago.

Para simular la iluminación existente en los pasillos y aulas mediante los tubos fluorescentes, la escalera principal con sus luces redondas y los talleres, se han utilizado puntos de luz con una tonalidad de luz blanca y un rango de amplitud corto, para que unas luces no solapen su iluminación con la de la luz contigua y evitar que se consuman recursos innecesariamente (*Ilustración 107*).



*Ilustración 107: Iluminación interior. Características del punto de luz*

Para los plafones situados en las escaleras y los rellanos, como la iluminación que proporcionan es mucho menor que la de un fluorescente y la luz está muy enfocada, se ha optado por utilizar un foco de luz, el cual se ha configurado con un ángulo de apertura muy amplio y un rango de emisión corto (*Ilustración 108*).



*Ilustración 108: Iluminación interior. Características del foco de luz*

### **4.2.4. Interacción**

La interacción de la visita virtual se ha creado mediante varios scripts, que son los encargados de gestionar la lógica de la visita a través del código escrito en ellos. Las dos partes en las que vamos a dividir esta sección son: controlador de personaje (interacción usuario-personaje) e interacción con el entorno.

### Controlador de personaje

Unity 3D trae por defecto un controlador de personaje para interacción a través de teclado y ratón mediante vista en primera persona, llamado “First Person Controller”. Este controlador trae implementadas las opciones básicas, pero ha sido necesario configurarlo adecuadamente para que la experiencia sea satisfactoria. A continuación se explicarán las funciones a tener en cuenta en el controlador para que el visitante pueda realizar el tour sin ningún problema:

#### Movimiento del personaje:

- Avanzar: Tecla W o flecha arriba.
- Retroceder: Tecla S o flecha abajo.
- Desplazamiento izquierda: Tecla A o flecha izquierda.
- Desplazamiento derecha: Tecla D o flecha derecha.
- Interacción: Tecla E.
- Velocidad de desplazamiento: Se ha reducido la velocidad de desplazamiento del personaje (avanzar, retroceder y movimiento lateral) ya que la velocidad por defecto era muy elevada.

#### Cámara:

- Movimiento: Desplazamiento con el ratón.
- Sensibilidad: Se ha reducido la sensibilidad de movimiento del ratón en los ejes X e Y ya que la velocidad configurada por defecto era muy elevada, dificultando el movimiento de la cámara.
- Near Camera: Se ha tenido que reducir el valor de esta propiedad ya que si el visitante se acercaba hasta estar en contacto con un objeto (a la altura de la cámara), este no se mostraba y la vista atravesaba el elemento.

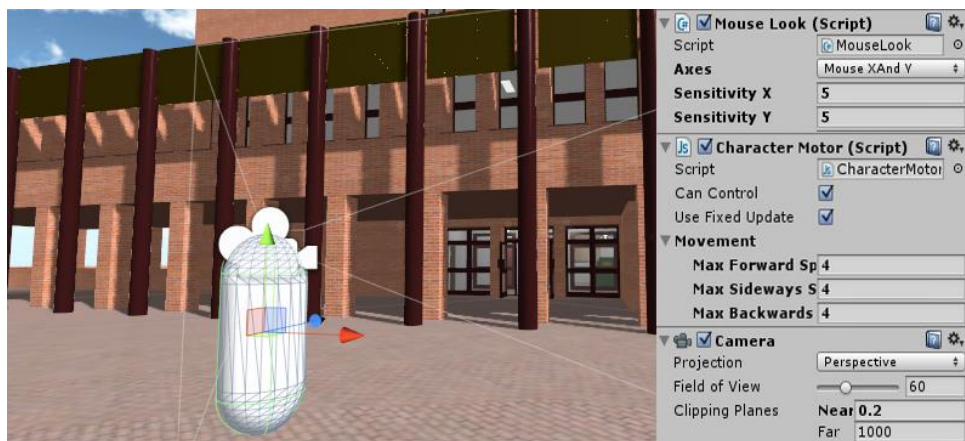


Ilustración 109: Propiedades del controlador de personaje

En la *Ilustración 109* se pueden observar las propiedades que se ha necesitado modificar en el controlador de personajes para que tanto la sensibilidad de movimiento a través del ratón, como la velocidad de movimiento, como las propiedades de proximidad de la cámara (Near camera) sean las idóneas para proporcionar el movimiento deseado.

### Interacción con entorno

El visitante, además de poder desplazarse por todo el edificio, tiene la posibilidad de interactuar con diferentes elementos del entorno para poder acceder a diferentes salas.

En esta visita, los objetos importantes con los que el visitante puede interactuar son las puertas. Por un lado están las puertas automáticas y por otro las puertas “dobles” (de color blanco las destinadas al acceso a las aulas de teoría y de ordenadores, y de color rojo las destinadas al acceso a los talleres).

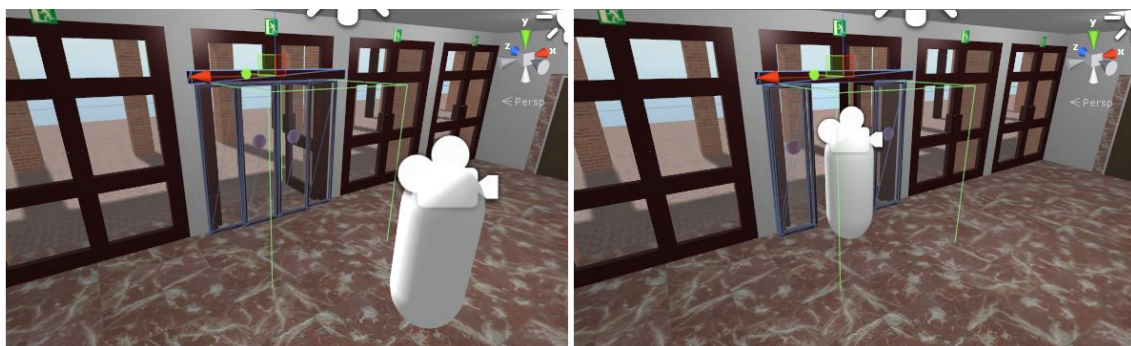
Para poder llevar a cabo esta tarea, es necesario que los objetos con los que se pretende realizar alguna acción posean el atributo collider (colisionador). Este atributo permite al motor de juego calcular si dos objetos están en contacto. Cualquier objeto sin un collider se comportará como un ente sin propiedades físicas, el cual se puede ver pero no se puede tocar, y por lo tanto al acercarse a él se le atraviesa.

Todos los elementos y modelos utilizados para la recreación de la visita poseen colliders para que el visitante no pueda atravesarlos, pero además en el caso de las puertas, se han añadido unos colliders especiales denominados triggers (disparadores). Por las características especiales de estos colliders, su función es la de comprobar si otro objeto entra, sale o se mantiene dentro de la zona delimitada por él mismo, y quedan reflejados mediante unas líneas verdes, como se podrá comprobar en siguientes ilustraciones donde se explica el funcionamiento de las puertas.

#### Puertas automáticas

La interacción con estas puertas es automática y la única acción que debe ejecutar el visitante es acercarse o alejarse, de la misma manera que ocurriría con unas puertas automáticas corrientes. Para el movimiento de las puertas se ha optado por configurar una animación predefinida que será explicada más adelante en la sección “4.2.5. Animaciones”, ya que al ser siempre el mismo movimiento, es más fácil realizarlo con esta técnica.

Para configurar estas puertas, el collider con propiedades de trigger comprueba si el visitante se encuentra dentro de una zona especial, y si es así, activa la animación correspondiente para abrir la puerta. Si por el contrario el visitante se aleja, la puerta se cerrará al detectar que se ha salido de la zona de interacción. En la *Ilustración 110* se puede observar gráficamente la reacción de la puerta según la proximidad del visitante. Para ampliar la información referente al script que se encarga de gestionar esta acción, puede dirigirse al “Anexo V: Scripts”.



**Ilustración 110: Interacción con puertas automáticas**

El collider que define la zona de acción de la puerta está definido por un cubo con líneas de color verde.

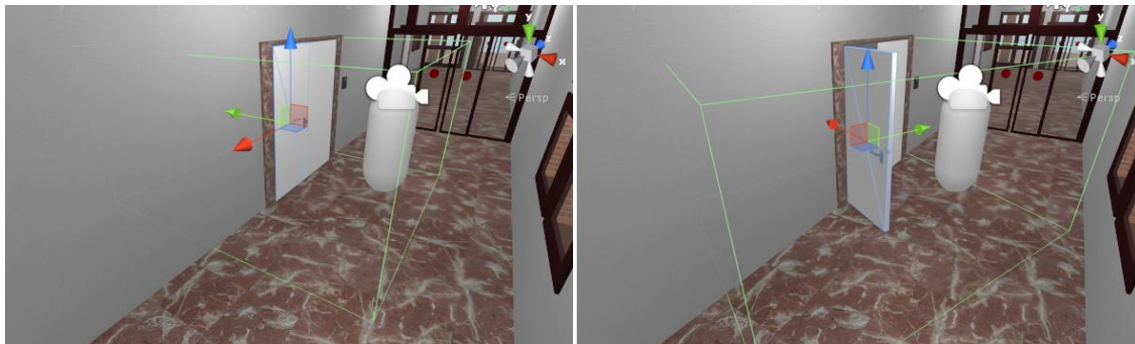


### Puertas dobles

En el caso de las puertas dobles, cuando el visitante se encuentra cerca de este elemento, es necesario que pulse la “tecla E” para indicar a la aplicación que se quiere abrir o cerrar la puerta. Se ha optado por configurar solamente las puertas dobles porque son las que permiten el acceso a las zonas comunes como aulas y talleres, mientras que las puertas individuales suelen estar posicionadas en el acceso a los baños y para los despachos.

De la misma manera que antes, es necesario configurar un collider adicional con propiedades de trigger para poder permitir calcular si el usuario está en una zona de acción próxima a la puerta, ya que no tendría sentido poder abrir puertas que se encuentran en el otro extremo del pasillo. A diferencia del script anterior, en este es necesaria la interacción directa por parte del visitante (pulsando la “tecla E”) para que se ejecute el movimiento de la puerta.

Además, se han configurado distintas propiedades de la puerta para que, aunque se utilice el mismo script en distintas puertas, cada una pueda reaccionar de una manera distinta. En la *Ilustración 111* se puede apreciar cómo es necesario que el visitante se encuentre dentro de la zona de acción de la puerta para poder interactuar con ella. Si se desea conocer con mayor detalle el proceso de creación del script para la apertura de puertas, puede dirigirse al “Anexo V: Scripts”.

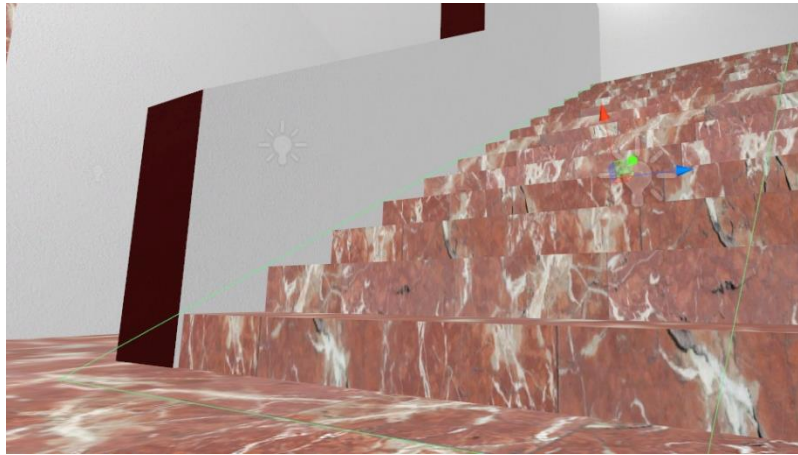


*Ilustración 111: Interacción con puertas de apertura manual*

El collider que define la zona de acción de la puerta está definido por un cubo con líneas de color verde.

### Escaleras

Al intentar desplazarse por las escaleras, el visitante se quedaba parado a mitad de trayecto. Además, se producía un efecto entre escalón y escalón similar al desplazamiento mediante saltos. Para evitar estos inconvenientes, se añadió una superficie plana transparente sobre las escaleras, generando una rampa por la que se desplazase el visitante de una manera fluida y sin los “saltos” comentados anteriormente. Los bordes de esta rampa transparente pueden verse en los laterales de la escalera mediante una fina línea de color verde.



*Ilustración 112: Rampa situada sobre las escaleras*

El objeto situado sobre las escaleras para crear una rampa está definido en color verde.

En la *Ilustración 112* se puede ver unas líneas de color verde a los lados de las escaleras que indican los bordes de la superficie transparente que se ha situado sobre los peldaños para permitir al visitante desplazarse sobre una rampa plana, de manera que no se produzcan movimientos extraños al cambiar de planta.

#### 4.2.5. Animaciones

A pesar de sólo haber realizado una animación (correspondiente a las puertas automáticas), se explicará el proceso de creación seguido para su realización mediante la herramienta “Animation” disponible en Unity 3D.

En esta herramienta, y mediante una línea de tiempo, configuramos las posiciones que queremos en nuestro objeto en cada momento, gestionando el propio motor el movimiento que debe realizar el objeto para llegar de una posición a la siguiente.

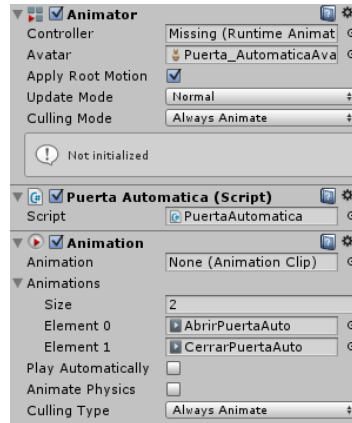
Como se observa en la *Ilustración 113*, nuestra animación sólo posee dos posiciones importantes (o keyframes), una al iniciar la animación y otra al terminar, al ser sólo necesaria la apertura lateral de las puertas. Dependiendo si la animación es para abrir o para cerrar la puerta, las posiciones en el tiempo de las puertas variarán.



*Ilustración 113: Ventana del componente Animation para crear animaciones*



Tras generar las dos animaciones, y habiendo creado anteriormente el script para la puerta automática, asignamos en el componente “Animation” de nuestra puerta las animaciones creadas, de tal manera que dejemos asignadas finalmente nuestras animaciones a la puerta, y gestionadas por el script “PuertaAutomatica”. En la *Ilustración 114* queda reflejada esta configuración.



*Ilustración 114: Configuración de las animaciones en las propiedades de la puerta*

#### 4.2.6. Menús

A continuación se van a mostrar los diseños finales de los menús. La aplicación de la visita virtual consta solamente de dos menús, el menú principal que aparece nada más arrancar la aplicación y el menú de pausa, accesible durante la visita pulsando la “tecla ESC”, los cuales van a ser explicados a continuación.

##### *Menú Principal*

El menú principal aparece nada más iniciar la aplicación, o cuando se solicita volver a él desde el menú de pausa pinchando sobre la opción “Menú Principal”.



*Ilustración 115: Menú Principal*

Como se puede observar en la *Ilustración 115*, la interfaz es muy sencilla, con sólo dos botones: uno para acceder a la visita virtual y otro para salir de la aplicación.

Si se quiere ampliar información sobre el script que gestiona las opciones del menú principal, puede dirigirse al “*Anexo V: Scripts*”.

### **Menú Pausa**

El menú de pausa se basa en el script “Pause Menu”<sup>4</sup> disponible en la Asset Store de Unity y modificado para ajustarse a las características de este proyecto. Contiene tres opciones: Menú Principal, Salir y Cambiar Calidad Gráfica (dentro de esta opción hay un submenú desplegable).

Para que el visitante pueda acceder a este menú debe estar realizando la visita y pulsar la “tecla ESC” (si la vuelve a pulsar, el visitante volverá a la visita), momento en el cual se bloqueará el movimiento del visitante y aparecerá sobrepuesto sobre su vista un menú con las opciones.

En la *Ilustración 116* se muestra la visita pausada, con el menú activo y dentro de la opción “Cambiar Calidad Gráfica”. Las opciones son autoexplicativas para seguir el esquema sencillo del menú principal y evitar las posibles dudas que puedan surgir a los visitantes al realizar el paseo virtual.



*Ilustración 116: Menú Pausa con el submenú desplegado*

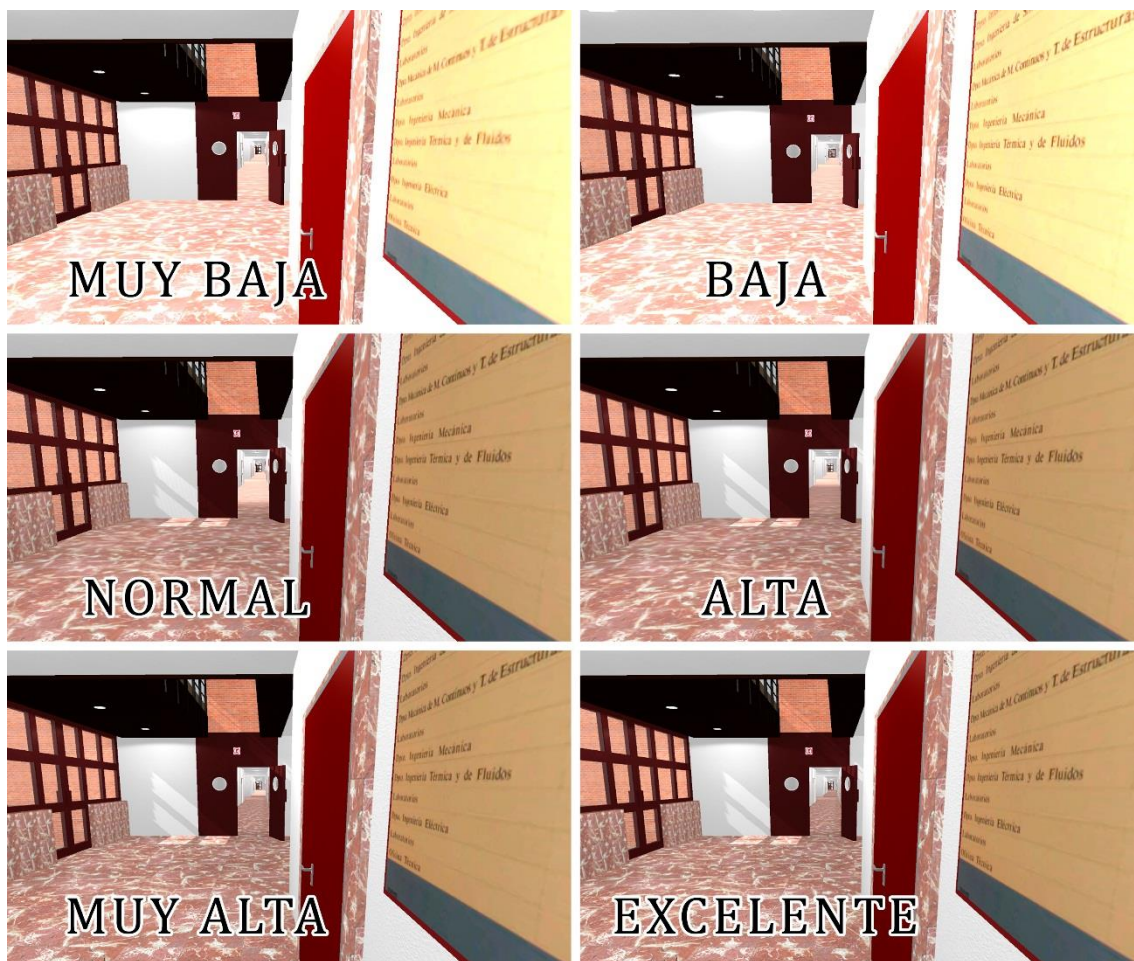
En la imagen que se encuentra en la siguiente página se pueden observar los cambios visuales entre cada opción de las disponibles en el submenú “Cambiar Calidad Gráfica”.

Si se configura la visita virtual en la calidad “Muy Baja”, la aplicación generará texturas de calidad inferior a la nativa y además no se mostrarán sombras, mientras que si se elige el nivel de calidad “Baja”, aunque la visita sigue sin renderizar sombras, ya se observan texturas de alta calidad (se puede comprobar en la nitidez de las letras del cartel situado a la derecha en la *Ilustración 117*).

<sup>4</sup> Disponible en: <https://www.assetstore.unity3d.com/en/#!/content/1020>

A partir de la siguiente calidad, denominada “Normal”, se mantienen las texturas de alta calidad y se empiezan a generar sombras (mejorando la calidad según se aumenta la opción de calidad gráfica), y lo más importante, la distancia de renderizado de alta calidad empieza a ampliarse.

Esto significa que cuanto mayor sea la calidad seleccionada, mayor es la distancia a la que el motor de juego gestiona que todos los componentes se vean tal y como fueron creados, mostrando las sombras y texturas de alta calidad. Este efecto se puede observar en la distancia hasta la que se generan las sombras y texturas en el pasillo principal de la imagen inferior, desde una distancia corta en calidad “Normal” hasta renderizar todos los elementos mostrados independientemente de su proximidad al visitante en la calidad “Excelente”.



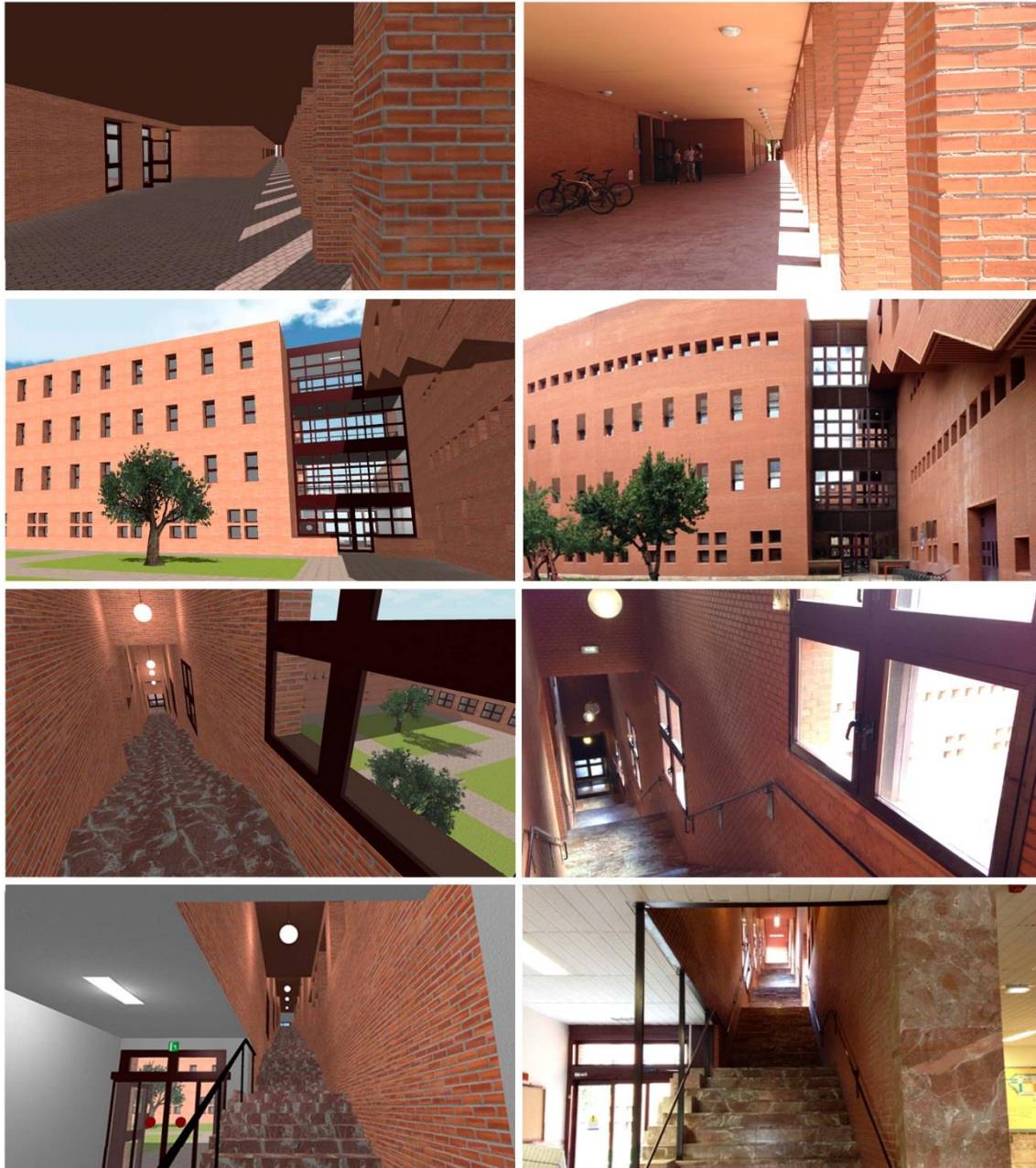
*Ilustración 117: Comparativa entre calidades gráficas*

Si se quiere profundizar en el código del script “PauseMenu”, la gestión de las distintas opciones del menú o de las diferentes calidades gráficas disponibles, se puede ampliar la información en el “Anexo V: Scripts”.



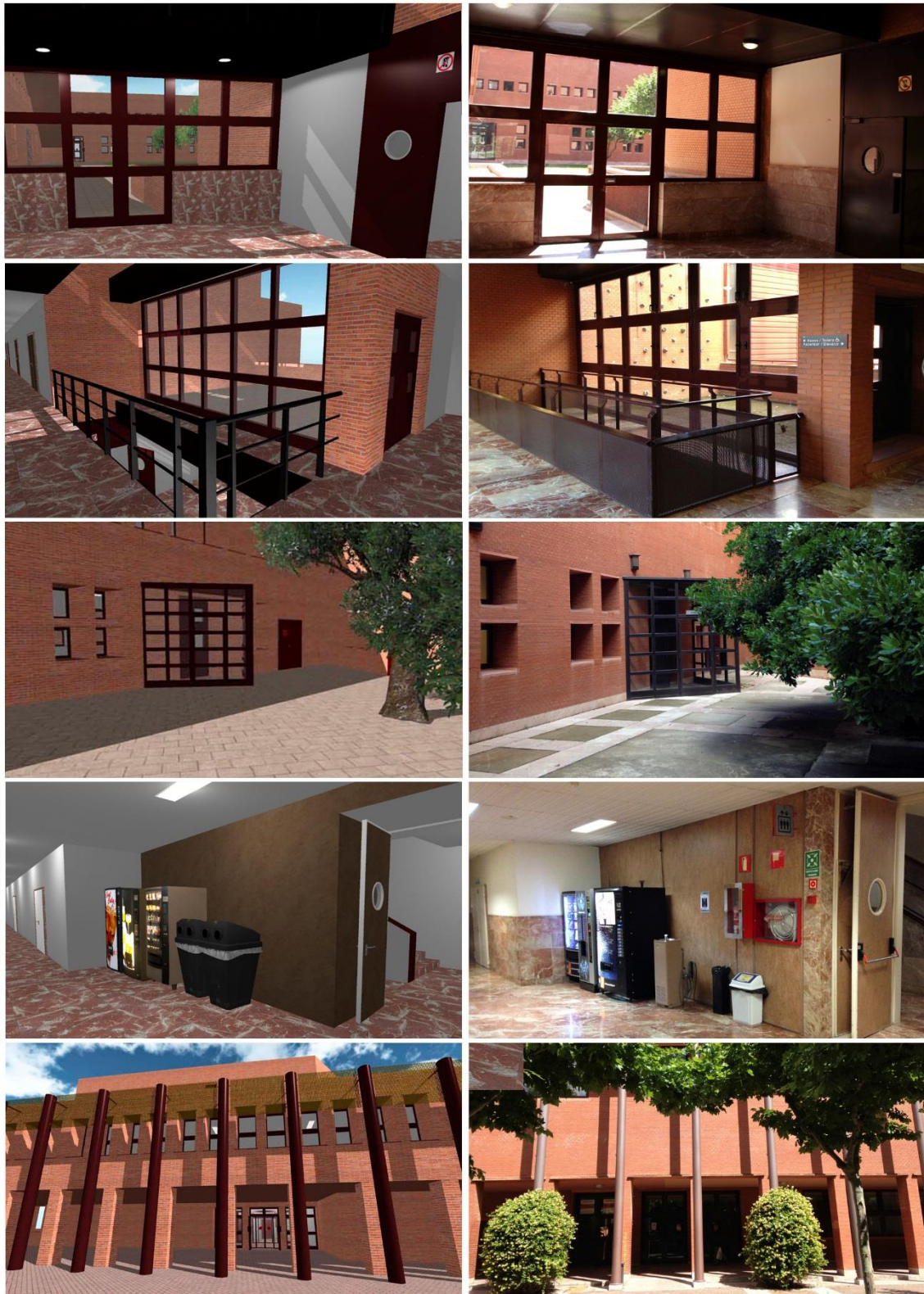
### 4.3. Resultado final

Después de haber explicado detalladamente todos los pasos seguidos para conseguir modelar e implementar los elementos necesarios para crear la visita virtual, se van a mostrar distintas imágenes (*Ilustración 118*, *Ilustración 119*, *Ilustración 120* e *Ilustración 121*) en las que se compara el resultado final con fotografías realizadas en el edificio Betancourt.



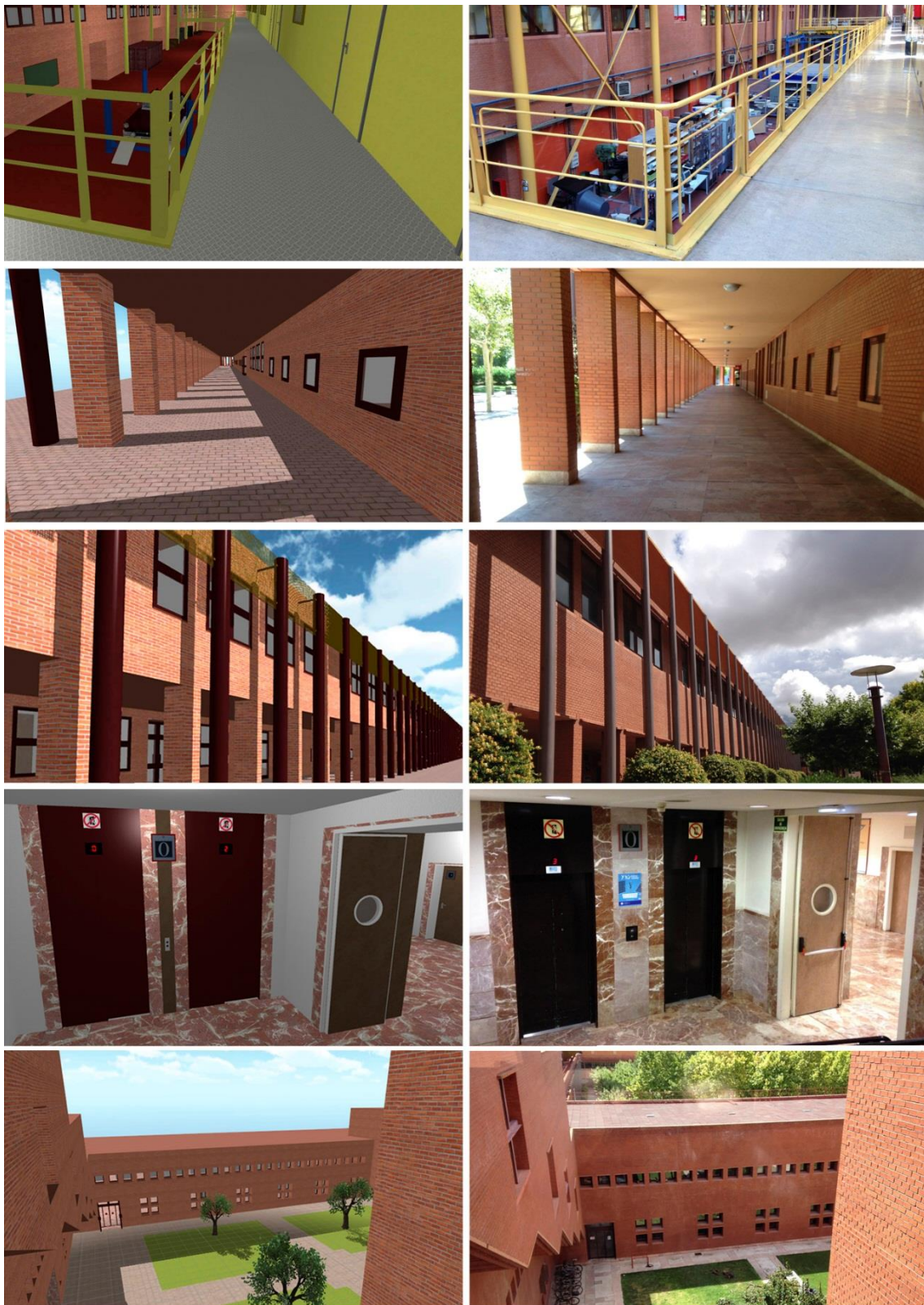
*Ilustración 118: Muestras del resultado final*





*Ilustración 119: Muestras del resultado final*





*Ilustración 120: Muestras del resultado final*





*Ilustración 121: Muestras del resultado final*

En este capítulo se ha explicado con profundidad y detenimiento todos los aspectos que componen la visita virtual, tanto el modelado de los objetos como su implementación para permitir la interacción del visitante en el edificio. Para comprobar que todo el trabajo realizado funciona correctamente es necesario someter a la aplicación a diferentes pruebas, las cuales serán explicadas en el siguiente apartado.

## 5. Pruebas

Después de haber explicado en el capítulo anterior el proceso de modelado 3D en Blender, y la implementación de los elementos diseñados y los sistemas de interacción en Unity 3D, es necesario probar la aplicación y comprobar si funciona correctamente.

A continuación se explicará el método empleado para probar la aplicación, así como los resultados obtenidos.

### 5.1. Pruebas realizadas y resultados obtenidos

Para comprobar que la aplicación funciona correctamente se han propuesto diferentes pruebas que han sido realizadas por personas ajenas al desarrollo del proyecto, indiferentemente si conocían con anterioridad el edificio Betancourt de la UC3M o no lo han visitado nunca.

Al grupo de 20 personas seleccionadas se le ha hecho entrega de un manual de usuario donde se explica el funcionamiento de la aplicación, las opciones disponibles durante la visita, y un cuestionario donde se detallan las pruebas que deben realizar para comprobar el correcto funcionamiento de la visita virtual en sus propios equipos (*Tabla 7*).

Los resultados obtenidos tras la realización de las pruebas por parte del equipo de pruebas pueden observarse a continuación en la *Tabla 8*.

### 5.2 Análisis de los resultados

Tras obtener los resultados de las pruebas, se comprueba que los resultados obtenidos son los esperados, por lo que se puede concluir que las pruebas han terminado satisfactoriamente.

Todos los participantes han sido capaces de ejecutar la aplicación sin ningún problema en sus equipos, sin la necesidad de instalar ningún software adicional, independientemente del equipo desde el que estuviesen trabajando, y con una funcionalidad total de todas las opciones disponibles.

Como sugerencias más importantes, aparte de las posibles mejoras en determinadas texturas o ampliación de diseño del edificio Betancourt al resto del campus, han sido las de añadir interacción a la visita mediante un sistema diferente al movimiento por teclas, para ayudar a las personas no habituadas con la interacción tipo FPS en la que se basa la visita.

Como primera opción se ha sugerido que se incluyese un sistema de movimiento mediante clics de ratón, en donde el usuario pulsase en la pantalla donde quiere ir y el visitante se desplazase hasta allí sin necesidad de utilizar las teclas W, A, S, D o las flechas de dirección, sin que esto influya en la libertad de movimiento de la cámara.

Como segundo método, se ha sugerido tener disponibles los mapas de la Universidad en pantalla y que el usuario pulsase en el lugar al que le gustaría desplazarse y que el visitante se moviese automáticamente hasta ese punto.

Estos nuevos sistemas de interacción son puntos interesantes a estudiar de cara a mejorar la experiencia y el acercamiento de la visita virtual a un público no habituado a utilizar el ordenador, ya que la finalidad de este proyecto es posibilitar a cualquier persona a conocer de una manera cómoda la Universidad, y para ello es necesario conseguir que su atención se centre en lo que ven en pantalla y no en cómo deben mover al visitante.

Pregunta		Respuesta
1	¿El ejecutable se inicia correctamente?	Si/No
2	¿Es necesaria alguna instalación adicional?	Si/No
3	¿El menú principal contiene las opciones “Vista Virtual” y “Salir”?	Si/No
4	En el menú principal, ¿funciona la opción “Visita Virtual”?	Si/No
5	En el menú principal, ¿funciona la opción de “Salir”?	Si/No
6	Durante la visita, ¿el visitante se desplaza al pulsar las flechas de dirección o las teclas W, A, S, D?	Si/No
7	Durante la visita, ¿la cámara se mueve si se desplaza el ratón?	Si/No
8	Durante la visita, ¿las puertas dobles se abren/cierran al pulsar la “tecla E”?	Si/No
9	Durante la visita, ¿las puertas automáticas se abren/cierran al acercarse/alejarse?	Si/No
10	Durante la visita, ¿accedes al menú de pausa si pulsas la “tecla ESC”?	Si/No
11	En el menú de pausa, ¿vuelves a la visita virtual si pulsas la “tecla ESC”?	Si/No
12	En el menú de pausa, si pulsas sobre la opción “Menú Principal”, ¿funciona correctamente?	Si/No
13	En el menú de pausa, si pulsas sobre la opción “Cambiar Calidad Gráfica”, ¿aparece un submenú con diferentes opciones gráficas?	Si/No
14	En el submenú “Cambiar Calidad Gráfica”, si pulsas sobre las diferentes opciones, ¿notas algún cambio visual?	Si/No
15	¿Le ha resultado útil la opción de poder cambiar la calidad gráfica para mejorar su experiencia al realizar la visita virtual?	Si/No
16	En el menú de pausa, si pulsas sobre la opción “Salir”, ¿se cierra la aplicación?	Si/No
17	¿Ha visitado alguna vez el edificio Agustín de Betancourt de la UC3M antes de realizar esta visita virtual?	Si/No
Si la respuesta a la pregunta 17 es afirmativa, responda a las preguntas 18 y 19. Si por el contrario la respuesta ha sido negativa, responda a la pregunta 20.		
18	¿Reconoce exteriormente el edificio Agustín de Betancourt?	Si/No
19	Si se desplaza por dentro del edificio, ¿reconoce los distintos lugares y pasillos?	Si/No
20	¿Cree que haber realizado la visita virtual le será de ayuda para ubicarse si decide realizar en el futuro una visita al campus?	Si/No
OBSERVACIONES		

Tabla 7: Cuestionario de evaluación de pruebas

Pregunta	Respuesta	
	SI	NO
1 ¿El ejecutable se inicia correctamente?	100%	0%
2 ¿Es necesaria alguna instalación adicional?	0%	100%
3 ¿El menú principal contiene las opciones “Vista Virtual” y “Salir”?	100%	0%
4 En el menú principal, ¿funciona la opción “Visita Virtual”?	100%	0%
5 En el menú principal, ¿funciona la opción de “Salir”?	100%	0%
6 Durante la visita, ¿el visitante se desplaza al pulsar las flechas de dirección o las teclas W, A, S, D?	100%	0%
7 Durante la visita, ¿la cámara se mueve si se desplaza el ratón?	100%	0%
8 Durante la visita, ¿las puertas dobles se abren/cierran al pulsar la “tecla E”?	100%	0%
9 Durante la visita, ¿las puertas automáticas se abren/cierran al acercarse/alejarse?	100%	0%
10 Durante la visita, ¿accedes al menú de pausa si pulsas la “tecla ESC”?	100%	0%
11 En el menú de pausa, ¿vuelves a la visita virtual si pulsas la “tecla ESC”?	100%	0%
12 En el menú de pausa, si pulsas sobre la opción “Menú Principal”, ¿funciona correctamente?	100%	0%
13 En el menú de pausa, si pulsas sobre la opción “Cambiar Calidad Gráfica”, ¿aparece un submenú con diferentes opciones gráficas?	100%	0%
14 En el submenú “Cambiar Calidad Gráfica”, si pulsas sobre las diferentes opciones, ¿notas algún cambio visual?	100%	0%
15 ¿Le ha resultado útil la opción de poder cambiar la calidad gráfica para mejorar su experiencia al realizar la visita virtual?	100%	0%
16 En el menú de pausa, si pulsas sobre la opción “Salir”, ¿se cierra la aplicación?	100%	0%
17 ¿Ha visitado alguna vez el edificio Agustín de Betancourt de la UC3M antes de realizar esta visita virtual?	60%	40%
Si la respuesta a la pregunta 17 es afirmativa, responda a las preguntas 18 y 19. Si por el contrario la respuesta ha sido negativa, responda a la pregunta 20.		
18 ¿Reconoce exteriormente el edificio Agustín de Betancourt?	100%	0%
19 Si se desplaza por dentro del edificio, ¿reconoce los distintos lugares y pasillos?	100%	0%
20 ¿Cree que haber realizado la visita virtual le será de ayuda para ubicarse si decide realizar en el futuro una visita al campus?	100%	0%
<b>OBSERVACIONES</b>		
<ul style="list-style-type: none"> <li>Añadir movimiento mediante clic de ratón. Pulsas en la pantalla donde quieres ir y el visitante se mueve solo.</li> <li>Añadir movimiento automático mediante interacción en un mapa.</li> </ul>		

Tabla 8: Resultados de las pruebas

En el siguiente apartado se explicará la planificación estimada y real del proyecto, así como el presupuesto en detalle elaborado para la creación del proyecto.

## 6. Planificación y presupuesto

---

Después de haber explicado en su totalidad el trabajo realizado para todo el proyecto, englobando desde los estudios iniciales hasta las pruebas realizadas, descritas en el punto anterior, queda por conocer la planificación y el presupuesto asignados a la realización de la visita virtual.

### 6.1. Planificación

Al inicio del proyecto se realizó una planificación estimada de las tareas que se debían realizar para completar el trabajo. Las tareas siguen la misma estructura en la que se ha basado este documento para explicar todo el proceso de creación de la visita virtual, como se puede observar a continuación:

- Estudio preliminar:
  - Estado del arte.
  - Herramientas de diseño 3D.
  - Objetivos.
- Desarrollo:
  - Análisis:
    - Requisitos del sistema.
    - Casos de uso.
  - Diseño:
    - Aplicación.
    - Menús.
    - Escenario.
- Implementación:
  - Modelado con Blender.
  - Implementación con Unity 3D.
- Pruebas.
- Documentación.

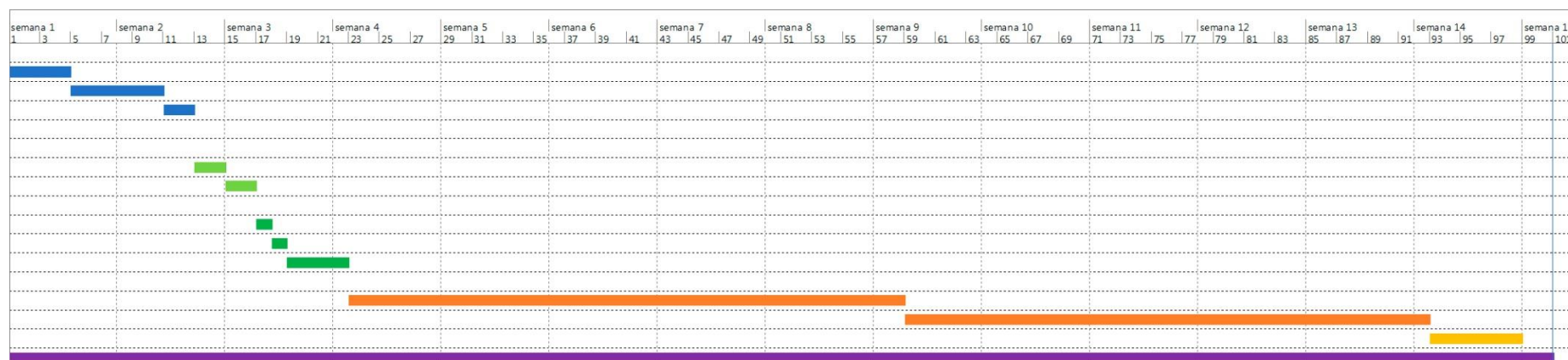


### 6.1.1. Estimada

Para la realización completa del proyecto se estimó inicialmente una duración de 100 días, repartiendo cada tarea según viene reflejado en la *Ilustración 122*. La duración estimada de cada tarea se puede apreciar en la *Ilustración 123*, donde se ha dividido la escala de tiempo en días y semanas.

Nombre de tarea	Inicio Estimado	Final Estimado	Duración Estimada
<b>Estudio Preliminar</b>	<b>1</b>	<b>12</b>	<b>12 días</b>
Estado del arte	1	4	4 días
Herramientas de diseño 3D	5	10	6 días
Objetivos	11	12	2 días
<b>Desarrollo</b>	<b>13</b>	<b>22</b>	<b>10 días</b>
<b>Análisis</b>	<b>14</b>	<b>16</b>	<b>4 días</b>
Requisitos del sistema	13	14	2 días
Casos de uso	15	16	2 días
<b>Diseño</b>	<b>17</b>	<b>22</b>	<b>6 días</b>
Aplicación	17	17	1 día
Menús	18	18	1 día
Escenario	19	22	4 días
<b>Implementación</b>	<b>23</b>	<b>92</b>	<b>70 días</b>
Modelado con Blender	23	58	36 días
Implementación con Unity 3D	59	92	34 días
<b>Pruebas</b>	<b>93</b>	<b>98</b>	<b>6 días</b>
<b>Documentación</b>	<b>1</b>	<b>100</b>	<b>100 días</b>

*Ilustración 122: Planificación estimada de tareas*



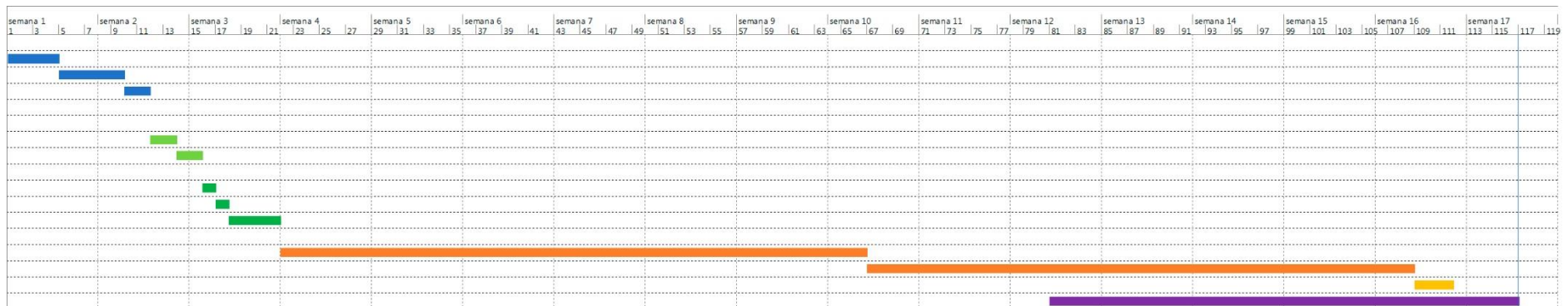
*Ilustración 123: Planificación estimada de trabajo*

### 6.1.2. Real

Aunque se estimaron 100 días para la realización completa del proyecto, la duración final ha ocupado 116 días. En la *Ilustración 124* se puede ver el tiempo real usado para terminar el trabajo, mientras que en la *Ilustración 125* se puede ver de una manera más gráfica la duración de cada tarea.

Nombre de tarea	Inicio Estimado	Final Estimado	Duración Estimada	Inicio Real	Final Real	Duración real
<b>Estudio Preliminar</b>	<b>1</b>	<b>12</b>	<b>12 días</b>	<b>1</b>	<b>11</b>	<b>11 días</b>
Estado del arte	1	4	4 días	1	4	4 días
Herramientas de diseño 3D	5	10	6 días	5	9	5 días
Objetivos	11	12	2 días	10	11	2 días
<b>Desarrollo</b>	<b>13</b>	<b>22</b>	<b>10 días</b>	<b>12</b>	<b>21</b>	<b>10 días</b>
<b>Análisis</b>	<b>14</b>	<b>16</b>	<b>4 días</b>	<b>12</b>	<b>15</b>	<b>4 días</b>
Requisitos del sistema	13	14	2 días	12	13	2 días
Casos de uso	15	16	2 días	14	15	2 días
<b>Diseño</b>	<b>17</b>	<b>22</b>	<b>6 días</b>	<b>16</b>	<b>21</b>	<b>6 días</b>
Aplicación	17	17	1 día	16	16	1 día
Menús	18	18	1 día	17	17	1 día
Escenario	19	22	4 días	18	21	4 días
<b>Implementación</b>	<b>23</b>	<b>92</b>	<b>70 días</b>	<b>22</b>	<b>108</b>	<b>87 días</b>
Modelado con Blender	23	58	45 días	22	66	45 días
Implementación con Unity 3D	59	92	42 días	67	108	42 días
<b>Pruebas</b>	<b>93</b>	<b>98</b>	<b>6 días</b>	<b>109</b>	<b>111</b>	<b>3 días</b>
<b>Documentación</b>	<b>1</b>	<b>100</b>	<b>100 días</b>	<b>97</b>	<b>116</b>	<b>20 días</b>

*Ilustración 124: Planificación real de tareas*



*Ilustración 125: Planificación real de trabajo*

### 6.1.3. Metodología

Para la realización de este proyecto se ha utilizado la metodología de desarrollo ágil Scrum, una de las más extendidas y utilizadas en el ámbito del desarrollo software, ya que permite que todo el equipo tenga conocimiento del estado del proyecto en reuniones periódicas al considerar que el proyecto puede sufrir variaciones en cualquier momento. De esta manera, se ofrecen soluciones para ir corrigiendo y perfeccionando el producto final de una manera mucho más rápida y eficaz.

Los conceptos que conforman la metodología Scrum y que se deben desarrollar son:

- **Product Owner:** Representa al cliente. Además se encarga de que el equipo trabaje de forma adecuada desde la perspectiva del negocio. En nuestro proyecto corresponde con la figura del jefe de proyecto.
- **ScrumMaster:** Es el encargado de eliminar los obstáculos que impiden que el equipo alcance los objetivos del sprint. Actúa como protección entre el equipo y las posibles influencias que los distraigan. En nuestro proyecto corresponde con la figura del jefe de proyecto.
- **Equipo de desarrollo:** El equipo de este proyecto está formado por un jefe de proyecto, un analista, un diseñador, un programador y un tester.
- **Product Backlog:** Documento de un documento de alto nivel donde se detallan los requisitos del proyecto. Estos requisitos están detallados en el apartado “Requisitos del sistema”.
- **Sprint Backlog:** Asignación de las tareas a realizar para una fecha concreta. Esta información está detallada en el apartado “6.1. Planificación”.
- **Reunión de planificación:** Se realiza al comienzo de cada sprint para decidir todo el trabajo que hay que realizar hasta la siguiente reunión de control.
- **Reunión diaria:** Reunión rápida diaria en la que se tratan tres conceptos básicos: trabajo realizado el día anterior, problemas surgidos y trabajo a realizar ese día. De esta manera el equipo mantiene conocimiento sobre el estado del trabajo de todos los compañeros.
- **Reunión de revisión:** Reunión en la que se revisa y analiza todo el trabajo realizado en cada sprint para poder tomar las decisiones adecuadas de cara al trabajo que queda por realizar.

### 6.2. Presupuesto

De la misma manera que al inicio del proyecto se realizó una estimación de las tareas a realizar para completar el proyecto, también se realizó el presupuesto asociado a dicha planificación. En el siguiente apartado se detallarán los costes relacionados con la gestión de personal, hardware y software que el proyecto ha generado.

### 6.2.1. Recursos Humanos

Para poder realizar un cálculo del coste por hora, se ha buscado el precio de los distintos puestos de trabajo en el portal “infojobs” y se ha realizado la media entre la menor y mayor tarifa. Los costes obtenidos son:

- Jefe de proyecto: 3.289 €/mes
- Analista: 2.160 €/mes
- Diseñador: 2.200 €/mes
- Programador: 2.500 €/mes
- Tester: 1.000 €/mes

Sin embargo, el trabajo realizado se ha medido en días, y no en meses como se han calculado los sueldos, por lo que para obtener una estimación correcta se ha considerado que cada mes posee una media de 22 días laborables, quedando los costes como se indica en la *Tabla 9*.

Puesto	Dedicación (nº de días)	Coste/Mes	Total
Jefe de proyecto	12	3.289	1.794,00 €
Analista	4	2.160	392,72 €
Diseñador	6	2.200	600,00 €
Programador	70	2.500	7954,54 €
Tester	6	1.000	272,72 €
<b>TOTAL</b>			<b>11.013,98 €</b>

*Tabla 9: Estimación del presupuesto de recursos humanos*

### 6.2.2. Recursos materiales

Para poder realizar el trabajo correctamente se necesita un equipo de gama alta, ya que la realización de trabajos de modelado requiere de altas prestaciones tanto de procesador como de computación de gráficos, de la misma manera que también es necesaria la adquisición del software empleado. Estos costes se indican en la *Tabla 10* y *Tabla 11* respectivamente.

#### Hardware

Descripción	Coste	% de uso	Dedicación	Depreciación	Coste imputable
PC Intel i7 - 930	1.450,00 €	100%	5	60	120,83 €
<b>TOTAL</b>					<b>120,83 €</b>

*Tabla 10: Estimación presupuesto para hardware*

#### Software

Descripción	Coste	% de uso	Dedicación	Depreciación	Coste imputable
Unity 3D*	1.140,00 €	100 %	3	12	285,00 €
Adobe Photoshop	926,86 €	100%	1	12	77,24 €
Windows 7	79,00 €	100%	3	12	19,75 €
Microsoft Office 2013	79,00 €	100%	1	12	6,58 €
<b>TOTAL</b>					<b>388,57 €</b>

*Tabla 11: Estimación presupuesto para software*

\*La licencia “Unity 3D Free” impide la comercialización de los proyectos realizados, por lo que es necesario adquirir la licencia “Unity 3D Pro”.

### 6.2.3. Costes totales

Para elaborar un presupuesto completo es necesario incluir un porcentaje de riesgo (que cubra los gastos generados en caso de un posible retraso en la finalización del trabajo) y un porcentaje de beneficios, para posteriormente incluir el IVA y así tener el presupuesto final del proyecto, como se muestra en la *Tabla 12*.

Descripción	Coste
Recursos Humanos	11.013,98 €
Amortización Hardware	120,83 €
Amortización Software	388,57 €
TOTAL costes directos	11.523,38 €
Riesgo (20%)	2.304,68 €
Beneficio (15%)	1.728,51 €
TOTAL sin IVA	15.556,57 €
IVA (21%)	3.266,88 €
TOTAL con IVA	<b>18.823,45 €</b>

*Tabla 12: Estimación total del presupuesto*

El presupuesto total del proyecto asciende a la cantidad de:

**18.823,45 €**

*Dieciocho mil ochocientos veintitrés euros con cuarenta y cinco céntimos.*

*Leganés, a 9 de febrero de 2015*



*Fdo. Miguel Bohada Arranz*

Tras haber finalizado el proyecto se ha comprobado que la planificación real ha sido superior a la estimada, por lo que los costes en personal han sido más elevados que los calculados inicialmente. Previendo esta posibilidad, se estimó un riesgo de un 20%, por lo que el sobrecoste para cubrir los gastos de personal ha sido inferior al margen de riesgo y no ha sido necesario utilizar los beneficios para cubrir los costes de desarrollo del proyecto.



## 7. Marco regulador y entorno socio-económico

### 7.1. Marco regulador

Este proyecto se basa en la realización de una visita virtual utilizando las técnicas de desarrollo de un videojuego, y dado que no se ha encontrado ninguna información referente al marco regulador de las visitas virtuales, la información proporcionada se centrará en el ámbito de los videojuegos.

La industria del videojuego es una industria en continuo crecimiento, y debido a la gran demanda de ocio electrónico, ha sido necesario establecer un sistema de normas y criterios que regulen el contenido disponible para los usuarios, gestionado por PEGI (sistema europeo de clasificación de videojuegos y software de entretenimiento) y la Asociación Española de Videojuegos - AEVI (conocida como aDeSe hasta 2014). [24] [25]

- **Código nacional de autorregulación:** El “código ético de autorregulación sobre el etiquetado y publicidad de los productos de software de entretenimiento” fue elaborado por la Asociación Española de Distribuidores y Editores de Software de Entretenimiento (aDeSe) en abril del 2001 con la voluntad de adecuar la comercialización de sus productos a la sensibilidad de la sociedad española y con el doble objetivo de informar y proteger al consumidor, y dotar de la máxima transparencia a los criterios de comercialización del sector.
- **Código europeo de autorregulación:** Respaldado por la Comisión Europea, publicado por la ISFE (Interactive Software Federation of Europe) y avalado por aDeSe/AEVI en España, establece el código PEGI (Pan European Game Information), encargado de clasificar el contenido de los videojuegos y demás tipo de software de entretenimiento, que entró en vigor en abril de 2003.

El sistema PEGI es un mecanismo de autorregulación diseñado por la propia industria del videojuego y software de entretenimiento para ofrecer a los compradores información orientativa sobre la edad adecuada de consumo de los productos que ofrecen según el contenido que incorporan. Se aplica en 25 países sin tener relación alguna con la Unión Europea, entre ellos en España. [26] [27]

El sistema está integrado por dos tipos de iconos descriptores, uno relativo a la edad recomendada y otro al contenido específico susceptible de análisis. El diseño de los logotipos informativos se basa en las luces de seguridad vial, haciendo más fácil su interpretación. Además, los sellos de edad van acompañados de términos pictográficos sobre el contenido del juego.



*Ilustración 126: Código PEGI*

Adicionalmente se ha creado una nueva etiqueta denominada PEGI OK en respuesta a los juegos online existentes en las páginas web. Si el juego lleva la etiqueta PEGI OK significa que el juego está adaptado a todos los grupos de jugadores de cualquier edad, ya que no existe ningún contenido potencialmente inapropiado. En la *Ilustración 126* se pueden observar las distintas etiquetas del código PEGI.

En referencia directa con los recursos usados en este proyecto, ninguno de ellos posee restricciones para su uso. Las texturas han sido extraídas desde diferentes repositorios en las que todas las texturas son gratuitas y libres de copyright, mientras que los recursos conseguidos tanto en la Asset Store, como en las bases de datos de modelos 3D, poseen licencias gratuitas para uso comercial.

## 7.2. Entorno socio-económico

De la misma manera que no hay información sobre el marco regulador de las visitas virtuales, tampoco existe información referente al aspecto económico que las engloba. Sin embargo, los videojuegos sí poseen un impacto fuerte en la sociedad, por lo que el estudio se centrará en ese aspecto. [28]

Día tras día queda patente la continua expansión de la industria de los videojuegos a nivel mundial, tanto en número de seguidores como en el aspecto económico, y esto es debido principalmente al acercamiento que en los últimos años ha buscado este sector en las personas que no eran seguidoras del ocio electrónico.

Mientras que a los jugadores habituales se les intenta sorprender con tecnologías potentes y nuevos sistemas de entretenimiento que mantengan su atención en las últimas novedades del mercado, a los jugadores ocasionales o no habituados con el género se les ha intentado atraer mediante sistemas de interacción de fácil uso y rápidos resultados, de tal manera que consigan guiar su atención poco a poco hacia el resto del mercado existente.

Este conjunto de estrategias de mercado ha permitido que la industria crezca imparablemente al diversificar los frentes en los que las empresas pueden centrar sus esfuerzos. Estos datos quedan patentes en las cifras reflejadas sobre el año 2014, donde el tamaño del mercado mundial fue de 23.188 millones de euros, un 3% más que el año anterior. Por su parte, España

sigue la misma tendencia de crecimiento, donde se movieron 763 millones de euros en el año 2014, un aumento del 31% respecto al año 2013.

Además, la previsión para el periodo 2015-2018 prevé que se crezca hasta los 890 millones de euros anuales en 2018, un crecimiento total del 16.71% respecto al año 2014, lo que equivale a un incremento de los beneficios anuales superiores al 4%. Estos datos reflejan que el crecimiento nacional estará por encima del internacional, que se estima en poco más del 3% anual en el mismo periodo.

Estos datos indican claramente que este proyecto sería viable desde el punto de vista social, ya que día a día más gente siente curiosidad por las nuevas tecnologías gracias a que la industria está apostando por abrirse a un público no habituado a este sector. De esta manera, las personas no verían con extrañeza las visitas virtuales inmersivas, consiguiendo que su uso se extendiese hasta convertirse en una opción prioritaria a la hora de visitar nuevos lugares, de igual manera que ahora se realiza mediante otras técnicas menos completas como puede ser Google StreetView.

## 8. Conclusiones y trabajos futuros

---

En este último capítulo del documento se van a explicar las conclusiones a las que se ha llegado tras la realización completa del proyecto, que consistía en el modelado tridimensional de una parte del campus de Leganés, centrado en el edificio Betancourt, para convertirlo mediante técnicas de desarrollo de videojuegos en una visita virtual tridimensional.

Este proyecto quería ofrecer al usuario una experiencia inmersiva dotada de libertad absoluta a la hora de moverse por el interior del edificio siendo él mismo quien decidiese qué partes quería examinar, ya que las visitas virtuales actuales se basan en un conjunto de fotografías panorámicas situadas en sitios predefinidos, pero que no llegan a aportar toda la información que pudiese desear un visitante, al ser las opciones muy limitadas

Tras la finalización del proyecto se puede concluir que todos los objetivos que se marcaron al inicio se han cumplido correctamente, y para apoyar esta afirmación se irán recordando los objetivos marcados al iniciar el desarrollo del proyecto, y cómo se han realizado. Posteriormente se comentarán las aportaciones realizadas al proyecto que no estaban previstas inicialmente, los problemas surgidos a lo largo de todo el proceso de creación, y las líneas futuras de desarrollo y trabajos que se podrían desarrollar partiendo del proyecto ya creado.

El trabajo se inició realizando un completo estudio del arte de las visitas virtuales en nuestra sociedad y la evolución que había sufrido la industria de los videojuegos desde su nacimiento hasta la actualidad, de tal manera que ayudase a comprender mejor el estado actual en el que se encontraban y cómo la unión de ambos sectores podrían ayudar a la sociedad.

Tomando como base las técnicas de desarrollo de los videojuegos se pretendía realizar una visita virtual interactiva donde el visitante tuviese una libertad absoluta de movimiento, por lo que fue necesario analizar las herramientas tanto de diseño 3D como motores de juego para concluir qué programas eran los más adecuados para realizar el proyecto que se pretendía llevar a cabo. De la misma manera, fue necesario estudiar los métodos de interacción existentes para decidir cuál sería el más apropiado para nuestra aplicación, permitiendo a un número máximo de personas acceder a la experiencia buscada con la visita.

El siguiente paso en los objetivos propuestos, fue estudiar las necesidades de la aplicación para poder desarrollar la visita. Era necesario diseñar y modelar multitud de objetos con Blender y posteriormente implementarlos con el motor de juego Unity 3D.

A nivel de modelado fue necesario crear el edificio Betancourt desde cero, después de haber realizado un profundo estudio de su estructura mediante planos, y documentación técnica de edificación y fotográfica, para poder conseguir un diseño realista. Además, también se crearon elementos de mobiliario y decoración para completar el edificio, como las puertas, ventanas, escaleras, etc. Por último, hizo falta dotar de realismo a todos estos modelos mediante diversas técnicas de texturización.

Después de realizar todos los modelos fue necesario implementarlos mediante el motor de juego para poder crear un escenario reconocible donde el usuario pueda sentir que visita realmente el campus. Para ayudar a completar el escenario se buscaron varios modelos externos y así conseguir una experiencia más inmersiva. Además, quedaba la tarea de implementar el método de interacción para que el usuario pudiese desplazarse por el edificio mediante teclado y ratón e interactuar con el entorno a través de las puertas.

Como mejoras al trabajo previsto inicialmente se decidió crear todas las estancias aunque no fuesen de acceso público, como los laboratorios, despachos o el aparcamiento subterráneo, y añadir diversos carteles de emergencias y evacuación para preparar la visita a posibles iniciativas futuras de simulacros de incendios o emergencias.

Tras realizar todo el trabajo de diseño e implementación, también se suministraron las planificaciones tanto estimadas como reales, para poder comparar el trabajo realizado y el cumplimiento de las fechas. De igual manera, se elaboró un presupuesto detallado con todos los conceptos necesarios para la realización del proyecto.

También está disponible el estudio realizado del entorno socio-económico en el que se engloba el trabajo, así como el cumplimiento del marco regulador que comprenden a los videojuegos y visitas virtuales.

Todo este trabajo realizado no ha estado exento de problemas, siendo necesario afrontar y buscar una solución para cada uno de ellos de tal manera que los objetivos marcados inicialmente se cumpliesen y el trabajo se completase en su totalidad.

El tiempo de creación de los modelos fue superior al previsto debido a que algunas de las técnicas empleadas para realizar modificaciones en los objetos no se podían utilizar directamente sobre el modelo ya creado, siendo necesario dividir cada uno en varios objetos diferentes antes de trabajar con ellos para que Blender pudiese realizar las operaciones correctamente.

Durante la implementación en Unity se detectó que trabajar con un número elevado de elementos perjudica el rendimiento, creando un renderizado sin fluidez, con saltos y lento. Para solucionar esto, como se estudió en el apartado “4.2.2. *Uso de recursos externos*”, es necesario utilizar la técnica “occlusion culling” (disponible solamente en la versión Pro), que limita el renderizado de objetos solamente a los elementos que se ven en la cámara, y no a todos los existentes en la escena.

Además, hay que mencionar la incompatibilidad ocasionada por algunos de los recursos externos a Unity (no entran en esta categoría los modelos propios creados en Blender y los recursos extraídos de la Asset Store), ya que muchos de estos objetos no cargaban las texturas e incluso se apreciaban fallos en el modelado, siendo necesaria su reparación si se querían usar.

Pero las conclusiones extraídas del desarrollo de este proyecto no se centran en analizar la realización de los objetivos iniciales, aportaciones extra efectuadas sobre el trabajo inicial o a la resolución de los problemas encontrados, sino que también se quiere dejar constancia que es un trabajo realizado con la intención de que en el futuro se pueda ampliar su alcance con mejoras o se utilice para distintos cometidos no previstos en su diseño inicial.



Como mejoras de la visita virtual, varias personas no habituadas al sistema de movimiento basado en FPS sugirieron implementar un nuevo método de interacción alternativo al uso del teclado y ratón, proponiendo como solución un movimiento basado en clics de ratón que indiquen al visitante el lugar hasta el que debe desplazarse.

Esta idea podría desarrollarse de varias maneras: La primera y más sencilla de las opciones sería una interacción directa en la que el visitante se desplazase directamente hasta el punto señalado con el ratón. Otra opción más elaborada podría ser la utilización de un mapa de apoyo y señalar en él la sala o el aula al que quiere desplazarse, implementando un sistema que detectase el camino idóneo hasta su destino.

Siguiendo con las mejoras de desplazamiento dentro del edificio, se podría implementar un sistema de localización de aulas similar al desplazamiento automático mediante un mapa, pero en el que simplemente se indicase el camino para llegar, simulando un navegador GPS. En este caso, el usuario sería el encargado de moverse siguiendo las indicaciones de la aplicación, pero decidiendo finalmente el camino que desea seguir para llegar a su destino.

Además, existe la posibilidad de añadir explicaciones de texto o auditivas para que ayudasen a entender mejor la Universidad, explicando la finalidad de cada laboratorio o taller.

También es importante no olvidarse de los nuevos métodos de interacción existentes en la actualidad, como las gafas de realidad virtual tipo Oculus Rift o cámaras con detección de movimiento tipo Kinect. Estos periféricos son más caros y es más difícil que el usuario medio tenga acceso a ellos, pero incrementarían notablemente la experiencia de usuario.

Pero durante la realización de este proyecto también se han estudiado las distintas líneas de trabajo futuro existentes, ya que no es un trabajo cerrado y puede utilizarse para múltiples utilidades. Estas ideas fueron surgiendo durante su desarrollo, pero se ha preferido no realizarlas porque, o diferían del enfoque que se le quería proporcionar, o era necesaria colaboración por parte de la Universidad, o el tiempo que se requería para implementarlas incrementaba considerablemente los plazos estimados para su finalización.

La primera línea de trabajo y posiblemente la más lógica, sería ampliar la visita virtual a todos los edificios del campus, de tal manera que los usuarios dispusiesen de una visión global de toda la Universidad y no solo estuviese centrada en el edificio Betancourt.

Para facilitar el acceso a la visita virtual, se podría implementar la aplicación en los navegadores web, ofreciendo en la propia página de la Universidad la posibilidad de realizar la visita virtual mediante fotografías o con la experiencia inmersiva que ofrece esta visita, evitando así que los usuarios necesitasen descargarse la aplicación en sus ordenadores.

Al ser un modelado realista, también se podría utilizar para estudiar métodos de evacuación ante emergencias, como podrían ser incendios. Sería necesario configurar el comportamiento que se desea para los alumnos y personal, pero ayudaría a mejorar la seguridad pudiendo prever el comportamiento de las personas según el nivel de peligro y su localización en el edificio.

Y como la línea de trabajo futuro más ambiciosa, estaría el desarrollo del proyecto en colaboración con toda la Universidad, incluyendo a la institución, los profesores, los

departamentos y los laboratorios. El trabajo final resultante consistiría en que todo el personal de la Universidad trabajase directamente en mejorar la visita.

Los profesores podrían tener en sus despachos los horarios disponibles de las tutorías, las notas de los cursos o diferentes avisos que les gustase comunicar al alumnado. Los departamentos, al acceder a su sala, podrían ofrecer toda la información referente a los proyectos y trabajos que realizan, servicios que ofrecen, o el acceso a su web. Los talleres y laboratorios podrían diseñar sus salas para que la gente supiese el equipamiento del que disponen, o mediante animaciones y videos mostrar sus avances en investigación.

De igual manera, los alumnos podrían acceder al material necesario para realizar sus prácticas. También serviría para la emisión de clases en streaming, donde al llegar al aula correspondiente se mostrase la información de la materia y el profesor que la está impartiendo, y acceder al contenido en directo.

Además, cualquier persona no familiarizada con el campus pero que necesitase ir y se le hubiese suministrado plaza de aparcamiento, podría saber con exactitud dónde se encuentra su plaza reservada y el itinerario a seguir para llegar a su ponencia o sala de reuniones a la que debiese acudir.

Por último, no debemos olvidar que esta visita virtual se la puede considerar como un videojuego educativo sin objetivos directos, por lo que se podría estudiar cómo gestionar determinados retos o niveles que ayudasen a los alumnos a conocer mejor la Universidad según fuesen avanzando en el desarrollo.

## 9. Bibliografía

---

- [1] «Universidad Carlos III de Madrid - Visita Virtual,» [En línea]. Available: [http://hosting01.uc3m.es/semanal3/visita\\_virtual/leganes\\_ext\\_es\\_en/leganes\\_ext.html](http://hosting01.uc3m.es/semanal3/visita_virtual/leganes_ext_es_en/leganes_ext.html).
- [2] «Facultad de Informática de Barcelona - Visita Virtual,» [En línea]. Available: <http://www.fib.upc.edu/es/centre/presentacio/visita.html>.
- [3] «Universidad de Jaén - Visita Virtual,» [En línea]. Available: <http://www.ujaen.es/visita-virtual-uj/en/>.
- [4] «Massachusetts Institute of Technology - Virtual Tour,» [En línea]. Available: <http://web.mit.edu/vrtour/>.
- [5] «Stanford University - Virtual Tours,» [En línea]. Available: <http://visit.stanford.edu/tours/virtual/>.
- [6] «Massachusetts Institute of Technology Virtual Tour on YouVisit,» [En línea]. Available: <https://www.youvisit.com/mit>.
- [7] S. Belli y C. López, «Breve historia de los videojuegos,» *Athenea Digital*, nº 14, pp. 159-179, 2008.
- [8] «Historia de los Videojuegos: EL Origen y los Inicios - Otakufreaks,» [En línea]. Available: <http://www.otakufreaks.com/historia-de-los-videojuegos-el-origen-y-los-inicios/>.
- [9] «Historia de los videojuegos - Facultad de Informática de Barcelona,» [En línea]. Available: <http://www.fib.upc.edu/retro-informatica/historia/videojocs.html>.
- [10] «Historia de los videojuegos - ElOtroLado,» [En línea]. Available: [http://www.elotrolado.net/wiki/Historia\\_de\\_los\\_videojuegos](http://www.elotrolado.net/wiki/Historia_de_los_videojuegos).
- [11] «Historia de los Videojuegos - Wikipedia,» [En línea]. Available: [http://es.wikipedia.org/wiki/Historia\\_de\\_los\\_videojuegos](http://es.wikipedia.org/wiki/Historia_de_los_videojuegos).
- [12] «Autodesk 3ds Max,» [En línea]. Available: <http://www.autodesk.es/products/3ds-max/overview>.
- [13] «Autodesk Maya,» [En línea]. Available: <http://www.autodesk.es/products/maya/overview>.
- [14] «Blender,» [En línea]. Available: <http://www.blender.org/>.
- [15] «CryEngine,» [En línea]. Available: <http://cryengine.com/>.
- [16] «Unreal Engine,» [En línea]. Available: <https://www.unrealengine.com/>.
- [17] «Unity,» [En línea]. Available: <https://unity3d.com/es>.
- [18] J. Duch i Gavalà y H. Tejedor Navarro, «Diseño de videojuegos: Sonido, interacción y redes,» UOC, 2013.
- [19] «Unidad Técnica de Mantenimiento y Obras UC3M,» [En línea]. Available: [http://portal.uc3m.es/portal/page/portal/unidad\\_tecnica\\_mantenimiento\\_obras/planos\\_tecnicos/leganes](http://portal.uc3m.es/portal/page/portal/unidad_tecnica_mantenimiento_obras/planos_tecnicos/leganes).
- [20] «Edificio Agustín de Betancourt - Google Maps,» [En línea]. Available: <https://goo.gl/maps/8lgsa>.

- [21] «Código Técnico de la Edificación - Ministerio de Fomento,» [En línea]. Available: <http://www.codigotecnico.org/web/recursos/documentos/>.
- [22] D. Vallejo et al, «Desarrollo de Videojuegos: Un Enfoque Práctico,» 2014.
- [23] «Unity - Manual: Formatos 3D,» [En línea]. Available: <http://docs.unity3d.com/es/current/Manual/3D-formats.html>.
- [24] «Agencia Española de Consumo, Seguridad Alimentaria y Nutrición - Informes y Documentos - Corregulación - Videojuegos,» [En línea]. Available: <http://consumo-inc.gob.es/informes/corregulacion/codigo2Etico.htm>.
- [25] aDeSe, «Dossier de Prensa PEGI,» Junio 2003. [En línea]. Available: [http://www.adese.es/pdf/Dossier\\_de\\_prensa\\_PEGI.pdf](http://www.adese.es/pdf/Dossier_de_prensa_PEGI.pdf).
- [26] «Asociación Española de Videojuegos,» [En línea]. Available: <http://www.aevi.org.es/index.php>.
- [27] «PEGI Pan European Game Information,» [En línea]. Available: <http://www.pegi.info/es/index/>.
- [28] «EAE Business School,» [En línea]. Available: <http://www.eae.es/news/2015/01/26/el-mercado-del-videojuego-en-espana-movio-763-millones-de-en-2014-con-un-crecimiento-del-31-respecto-al-2013>.

## Anexo I: Acrónimos

---

**3DS:** Formato de archivo usado por programas de simulación y gráficos 3D como 3ds Max.

**aDeSe:** Asociación Española de Distribuidores y Editores de Software de Entretenimiento. En 2014 pasa a refundarse con el nombre de AEVI.

**AEVI:** Asociación Española de Videojuegos. Asociación que aspira a aglutinar a todos los agentes implicados en la cadena de valor del videojuego (desarrolladores, editores, comercializadores, etc.) en una asociación común que nace con el firme propósito de fortalecer y defender los intereses de las industrias del videojuego en España.

**BLEND:** Formato utilizado por Blender para guardar los proyectos creados con este software.

**CS:** Extensión de archivo que contiene código en lenguaje de programación C#.

**CTE:** Código Técnico de la Edificación.

**DB-DI:** Documento Básico de Seguridad en caso de Incendio.

**DB-SUA:** Documento Básico de Seguridad de Utilización y Accesibilidad.

**FBX:** Formato de archivo 3D independiente de la plataforma, que proporciona acceso al contenido creado en cualquier paquete de software.

**FPS:** First Person Shooter. Género de videojuego de disparos con vista en primera persona.

**JPG/JPEG:** Joint Photographic Experts Group. Formato gráfico que utiliza un algoritmo de compresión con pérdida para reducir el tamaño de los archivos de imágenes.

**JS:** Extensión de archivo que contiene código en lenguaje de programación JavaScript.

**MMORPG:** Massively multiplayer online role-playing game. Género de videojuego de rol que permite la interacción simultánea en mundos virtuales de un gran número de jugadores.

**PDF:** Portable Document Format. Formato de almacenamiento para documentos digitales.

**PEGI:** Pan European Game Information. Sistema de clasificación por edades del contenido de videojuegos y ocio electrónico.

**PNG:** Portable Network Graphics. Formato gráfico para guardar imágenes basado en un algoritmo de compresión sin pérdida de bitmaps.

**RSF:** Requisito de software funcional.

**RSNF:** Requisito de software no funcional.

**UC3M:** Universidad Carlos III de Madrid.

**VR:** Realidad Virtual.



## Anexo II: Glosario de términos

---

**3ds Max:** Software de modelado, animación y renderizado 3D desarrollado por Autodesk.

**Adobe Photoshop:** Software de edición de imágenes desarrollado por Adobe.

**Asset:** Recurso “externo” al motor gráfico.

**Blender:** Software de modelado y animación tridimensional desarrollado por Blender Foundation bajo licencia Open Source.

**CryEngine:** Motor de juego desarrollado por la Crytek.

**DirectX:** Librería para el tratamiento de gráficos en la Microsoft Windows.

**Game Engine:** Término anglosajón para referirse al motor de juego.

**GIMP:** Herramienta de edición de imágenes con licencia de software libre.

**JavaScript:** Lenguaje de programación orientado a objetos.

**Mapping:** O mapeado, técnica de aplicación de texturas en un modelo tridimensional.

**Maya:** Software de modelado, animación y simulación 3D desarrollado por Autodesk.

**Motor de juego:** Software de diseño y creación de videojuegos.

**OpenGL:** Open Graphics Library. Librería para el tratamiento de gráficos.

**Python:** Lenguaje de programación con énfasis en una sintaxis que favorezca un código legible.

**Plug-in:** También conocido como complemento, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

**Renderizar:** Proceso de generación de imágenes o video mediante el cálculo de iluminación partiendo de modelos 3D.

**Script:** Conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución.

**Scripting:** Técnica de creación de scripts en un entorno de desarrollo específico.

**Shader:** Unidad escrita en lenguaje de sombreado que se puede compilar independientemente.

**Streaming:** Distribución digital multimedia a través de la red donde se consume un producto en paralelo a su descarga.

**Unity 3D:** Motor de juego desarrollado por Unity Technologies.

**Unreal Engine:** Motor de juego desarrollado por Epic Games.

## Anexo III: Requisitos del sistema

En este anexo se detallan los requisitos de sistema, para poder conocer en profundidad todos los atributos que los componen.

Para poder clasificar correctamente cada requisito de sistema, se ha usado una plantilla estándar, la cual será explicada a continuación:

Identificador	
Título	
Descripción	
Prioridad	
Claridad	
Estabilidad	
Verificabilidad	
Necesidad	
Fuente	

*Tabla 13: Plantilla para requisitos funcionales y requisitos no funcionales*

Donde cada campo significa:

- **Identificador:** Código único asignado a cada requisito para su identificación.
- **Título:** Nombre breve y descriptivo del requisito.
- **Descripción:** Explicación breve, clara y concisa del requisito.
- **Prioridad:** Nivel de importancia del requisito. Cuanta mayor sea la prioridad, mayor será la necesidad de realizarlo.
- **Claridad:** Nivel de precisión del requisito. Cuanta menos claridad, mayor ambigüedad y posibilidad de confusión y malinterpretación.
- **Estabilidad:** Nivel de probabilidad de que el requisito no cambie. Cuanta mayor estabilidad, la probabilidad de que el requisito cambie es menor.
- **Verificabilidad:** Facilidad de comprobar la correcta aplicación del requisito.
- **Necesidad:** Nivel de importancia para el cliente.
- **Fuente:** Origen del requisito.

## Requisitos software funcionales

Identificador	RSF-01
Título	Menú Principal
Descripción	La aplicación deberá poseer un menú principal desde el que se pueda acceder a la visita virtual o salir de la aplicación
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 14: RSF-01 - Menú Principal*

Identificador	RSF-02
Título	Sistema de juego FPS
Descripción	El tipo de juego corresponderá con un videojuego de género FPS (First Person Shooter), del que se usará el tipo de interacción de movimientos y cámara
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 15: RSF-02 - Sistema de juego FPS*

Identificador	RSF-03
Título	Escenario
Descripción	El escenario mostrado debe corresponder con el edificio Agustín de Betancourt, perteneciente al campus de Leganés de la Universidad Carlos III de Madrid
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 16: RSF-03 - Escenario*

Identificador	RSF-04
Título	Interacción
Descripción	El usuario podrá moverse mediante el uso del teclado y el ratón
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 17: RSF-04 - Interacción*

<b>Identificador</b>	<b>RSF-05</b>
<b>Título</b>	Menú Pausa
<b>Descripción</b>	La aplicación deberá poseer un menú de pausa al que se podrá acceder en cualquier momento de la visita pulsando la "tecla ESC". Desde ese menú se podrá volver al menú principal, cambiar opciones gráficas o salir de la aplicación.
<b>Prioridad</b>	Alta
<b>Claridad</b>	Alta
<b>Estabilidad</b>	Alta
<b>Verificabilidad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Fuente</b>	Tutor

*Tabla 18: RSF-05 - Menú Pausa*

<b>Identificador</b>	<b>RSF-06</b>
<b>Título</b>	Calidad Gráfica
<b>Descripción</b>	La aplicación permitirá al usuario cambiar la calidad gráfica del juego en cualquier momento durante el transcurso de la visita
<b>Prioridad</b>	Baja
<b>Claridad</b>	Alta
<b>Estabilidad</b>	Media
<b>Verificabilidad</b>	Alta
<b>Necesidad</b>	Opcional
<b>Fuente</b>	Tutor

*Tabla 19: RSF-06 - Calidad Gráfica*

<b>Identificador</b>	<b>RSF-07</b>
<b>Título</b>	Salir Aplicación
<b>Descripción</b>	La aplicación permitirá salir de la aplicación al usuario en cualquier momento que lo desee
<b>Prioridad</b>	Alta
<b>Claridad</b>	Alta
<b>Estabilidad</b>	Alta
<b>Verificabilidad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Fuente</b>	Tutor

*Tabla 20: RSF-07 - Salir Aplicación*

## Requisitos software no funcionales

Identificador	RSNF-01
Título	Multiplataforma
Descripción	La aplicación deberá ser compatible con distintos Sistemas Operativos
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 21: RSNF-01 - Multiplataforma*

Identificador	RSNF-02
Título	Compatibilidad con resoluciones
Descripción	La aplicación deberá ser compatible con distintas resoluciones de pantalla
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 22: RSNF-02 - Compatibilidad con resoluciones*

Identificador	RSNF-03
Título	Fichero ejecutable
Descripción	La aplicación deberá ser un archivo ejecutable para simplificar el acceso a ella
Prioridad	Alta
Claridad	Alta
Estabilidad	Alta
Verificabilidad	Alta
Necesidad	Esencial
Fuente	Tutor

*Tabla 23: RSNF-03 - Fichero ejecutable*



## Anexo IV: Casos de uso

En este anexo se detallan los casos de uso, para poder conocer en profundidad todos los atributos que los componen.

Para poder clasificar correctamente cada requisito de sistema, se ha usado una plantilla estándar, la cual será explicada a continuación:

Identificador	
Título	
Descripción	
Actor	
Precondiciones	
Postcondiciones	
Escenario	

*Tabla 24: Plantilla para casos de uso*

Donde cada campo significa:

- **Identificador:** Código único asignado a cada requisito para su identificación.
- **Título:** Nombre breve y descriptivo del requisito.
- **Descripción:** Explicación breve, clara y concisa del requisito.
- **Actor:** Tipo de usuario (corresponde siempre con visitante).
- **Precondiciones:** Condiciones que deben cumplirse antes de ejecutar el caso de uso.
- **Postcondiciones:** Condiciones que debe cumplir el sistema al finalizar el caso de uso.
- **Escenario:** Etapas del caso de uso.

Además, se muestran ilustraciones que indican los casos de uso que puede realizar el actor visitante dependiendo de la pantalla en la que se encuentre.

### Pantalla: Menú Principal

El menú principal contiene las siguientes opciones:

- **Visita Virtual:** Inicia la visita virtual al edificio Agustín de Betancourt.
- **Salir:** Cierra la aplicación.

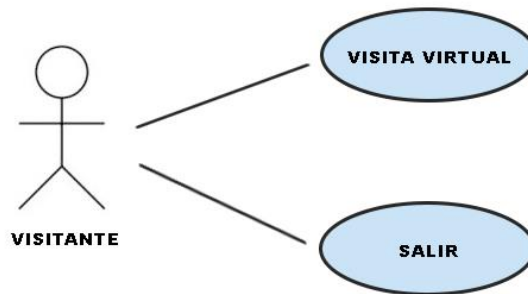


Ilustración 127: Casos de uso en "Menú Principal"

Identificador	CU-01
Título	Visita Virtual
Descripción	El visitante tiene como opción acceder a la visita virtual
Actor	Visitante
Precondiciones	Ninguna
Postcondiciones	Inicia la visita virtual
Escenario	El visitante inicia la aplicación y accede al menú principal, donde, tras elegir la opción "Visita Virtual" se iniciará la visita al edificio Agustín de Betancourt

Tabla 25: CU-01 - Visita Virtual

Identificador	CU-02
Título	Salir desde Menú Principal
Descripción	El visitante tiene como opción salir de la aplicación
Actor	Visitante
Precondiciones	La aplicación debe estar iniciada
Postcondiciones	Se cierra la aplicación
Escenario	El visitante inicia la aplicación y accede al menú principal, donde, tras elegir la opción "Salir", se cierra la aplicación

Tabla 26: CU-02 - Salir desde Menú Principal

### Pantalla: Visita Virtual

Mientras el visitante está realizando la visita virtual, puede realizar las siguientes acciones:

- **Mover Cámara:** Permite al visitante mover la visión de la cámara.
- **Mover Personaje:** Permite al visitante desplazarse por el entorno.
- **Abrir/Cerrar Puerta:** Permite al visitante abrir/cerrar determinadas puertas.

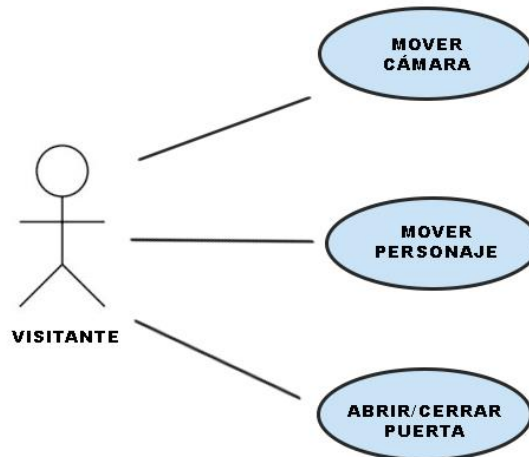


Ilustración 128: Casos de uso durante la "Visita Virtual"

Identificador	CU-03
Título	Mover cámara
Descripción	El visitante puede mover la cámara moviendo el ratón
Actor	Visitante
Precondiciones	La aplicación debe estar iniciada y haber accedido a la "Visita Virtual"
Postcondiciones	La visión mostrada por la cámara varía según el movimiento del ratón
Escenario	El visitante, tras acceder a la "Visita Virtual" desde el menú principal, tiene la posibilidad de modificar la vista mostrada según el movimiento de la cámara en ángulos de 360º

Tabla 27: CU-03 - Mover Cámara

Identificador	CU-04
Título	Mover personaje
Descripción	El visitante puede mover al personaje usando las teclas de dirección o las teclas W, A, S, D del teclado
Actor	Visitante
Precondiciones	La aplicación debe estar iniciada y haber accedido a la "Visita Virtual"
Postcondiciones	La visión mostrada por la cámara varía según las acciones indicada por el usuario mediante las teclas de movimiento
Escenario	El visitante, tras acceder a la "Visita Virtual" desde el menú principal, tiene la posibilidad de desplazarse por el entorno

Tabla 28: CU-04 - Mover Personaje

Identificador	CU-05
Título	Abrir/Cerrar Puerta
Descripción	El visitante podrá abrir/cerrar determinadas puertas pulsando la "tecla E"
Actor	Visitante
Precondiciones	La aplicación debe estar iniciada, haber accedido a la "Visita Virtual" y encontrarse el personaje dentro de la zona de interacción de la puerta elegida
Postcondiciones	La puerta se abre/cierra a voluntad del visitante
Escenario	El visitante, tras acceder a la "Visita Virtual" desde el menú principal, tiene la posibilidad de abrir y cerrar determinadas puertas a su elección

Tabla 29: CU-05 - Abrir/Cerrar Puerta

### Pantalla: Menú Pausa

Si el visitante accede al menú de pausa durante la visita, se mostrarán las siguientes opciones:

- **Menú Principal:** Permite al visitante acceder al menú principal.
- **Cambiar Calidad Gráfica:** Permite al visitante modificar la calidad gráfica de la visita virtual.
- **Salir:** Cierra la aplicación.

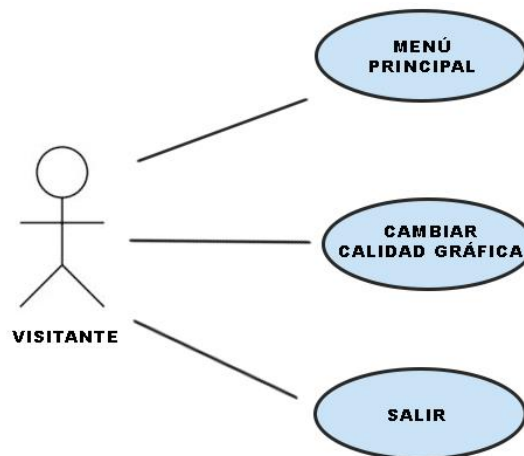


Ilustración 129: Casos de uso en "Menú Pausa"

Identificador	CU-06
Título	Ir a Menú principal
Descripción	Desde el menú de pausa, el visitante podrá volver a la pantalla de inicio
Actor	Visitante
Precondiciones	La aplicación debe estar abierta, iniciada la "Visita Virtual" y haber accedido al menú de "Pausa"
Postcondiciones	La aplicación sitúa al visitante en el "Menú Principal"
Escenario	El visitante, tras acceder al menú de Pausa desde la "Visita Virtual", tiene la posibilidad de volver al "Menú Principal"

Tabla 30: CU-06 - Ir a Menú Principal

<b>Identificador</b>	<b>CU-07</b>
<b>Título</b>	Cambiar Calidad Gráfica
<b>Descripción</b>	Desde el menú de pausa, el visitante podrá modificar la calidad gráfica de la "Visita Virtual"
<b>Actor</b>	Visitante
<b>Precondiciones</b>	La aplicación debe estar abierta, iniciada la "Visita Virtual" y haber accedido al menú de "Pausa"
<b>Postcondiciones</b>	Se muestra un submenú con las distintas opciones de calidad gráfica
<b>Escenario</b>	El visitante, tras acceder al menú "Pausa" desde el modo "Visita Virtual", tiene la posibilidad de elegir entre seis resoluciones distintas para adaptar la calidad gráfica de la visita a las características técnicas de su equipo

*Tabla 31: CU-07 - Cambiar Calidad Gráfica*

<b>Identificador</b>	<b>CU-08</b>
<b>Título</b>	Salir desde Menú Pausa
<b>Descripción</b>	Desde el menú de pausa, el visitante podrá salir de la aplicación
<b>Actor</b>	Visitante
<b>Precondiciones</b>	La aplicación debe estar abierta, iniciada la "Visita Virtual" y haber accedido al menú de "Pausa"
<b>Postcondiciones</b>	Se cierra la aplicación
<b>Escenario</b>	El visitante, tras acceder al menú "Pausa" desde el modo "Visita Virtual", tiene la posibilidad de cerrar la aplicación

*Tabla 32: CU-08 - Salir desde Menú Pausa*



# Anexo V: Scripts

En este anexo se van a explicar con mayor profundidad los diferentes scripts utilizados en este proyecto. Existen dos tipos de scripts, los de interacción con el entorno y los de gestión de pantallas y menús.

## *Puertas automáticas*

Para crear la interacción con las puertas automáticas se ha generado un script que permite abrir y cerrar la puerta según la proximidad del visitante a ellas. Para configurar estas puertas, el collider con propiedades de trigger comprueba si el visitante se encuentra dentro de una zona especial, y si es así, activa la animación correspondiente:

- Si el visitante entra en la zona de acción del collider-trigger (OnTriggerEnter), la puerta se abrirá llamando a la animación “AbrirPuertaAuto”.
- Si el visitante sale de la zona de acción del collider-trigger (OnTriggerExit), la puerta se cerrará ejecutando la animación “CerrarPuertaAuto”.

```
void OnTriggerEnter (Collider other) {
    animation.CrossFade("AbrirPuertaAuto");
}

void OnTriggerExit (Collider other){
    animation.CrossFade("CerrarPuertaAuto");
}
```

## *Puertas dobles*

En el caso de las puertas dobles, se ha buscado una interacción directa entre el visitante y la aplicación, siendo necesario que pulse la “tecla E” para indicar que tiene intención de abrir o cerrar la puerta. De la misma manera que antes, ha sido necesario utilizar un collider con propiedades de trigger para que reconozca si el usuario se encuentra dentro o fuera del área de acción.

Como se puede apreciar en el siguiente código, el script posee varias propiedades como son la velocidad de giro (\_velocidadGiro), el ángulo máximo de apertura (\_anguloAbierto), y el ángulo de cierre (\_anguloCerrado).

Para configurar el funcionamiento del script, se necesita que el collider-trigger compruebe que el visitante se encuentra dentro de su zona de acción, y si es así, estará disponibles la opción de abrir y cerrar.

Si el visitante está dentro de la zona de acción y pulsa la “tecla E”, llama a la rutina “Abrir” si se encuentra cerrada, o a la rutina “Cerrar” si por el contrario ya se encuentra abierta. Tanto “Abrir” como “Cerrar” realizan una rotación de la puerta respecto del eje Y con la velocidad definida en “\_velocidadGiro” hasta el ángulo correspondiente de apertura o cierre (según la acción que se esté llevando a cabo).

Las propiedades configuradas son:

```
bool _abierto = false;
public float _velocidadGiro = 1;
public float _anguloAbierto = 90;
public float _anguloCerrado = 0;
```

Si el visitante se encuentra dentro de la zona de acción de la puerta y pulsa la “tecla E” se comprueba si está abierta o cerrada y llama a la rutina correspondiente:

```
void OnTriggerStay (Collider other){
    if(Input.GetKeyUp(KeyCode.E)){
        if(_abierto)
            StartCoroutine(Cerrar());
        else
            StartCoroutine(Abrir());
    }
}
```

Las rutinas realizan la operación de giro en la puerta rotando respecto del eje Y (vertical) a la velocidad indicada hasta la nueva posición:

```
IEnumerator Abrir(){
    _abierto = true;
    while(transform.parent.localEulerAngles.y <= _anguloAbierto){
        transform.parent.Rotate(0, _velocidadGiro * Time.deltaTime, 0);
        yield return new WaitForEndOfFrame();
    }
}
IEnumerator Cerrar(){
    //Debug.Log("Cerrar");
    _abierto = false;
    while(transform.parent.localEulerAngles.y >= _anguloCerrado){
        transform.parent.Rotate(0, -_velocidadGiro * Time.deltaTime, 0);
        yield return new WaitForEndOfFrame();
    }
}
```

### **Menú Principal**

El menú principal aparece nada más iniciar la aplicación, o cuando se solicita volver a él desde el menú de pausa pinchando sobre la opción “Menú Principal”.

Posee un sencillo script llamado “LoadLevel” que gestiona las opciones de los dos botones.

- Si se pulsa sobre el botón “Visita Virtual”, el script ejecuta `changeToScene` y se carga la visita virtual (en nuestro caso, la visita corresponde con el nivel “Betancourt”).
- Si se pulsa sobre el botón “Salir”, el script ejecuta la instrucción `Application.Quit` y cierra la aplicación.

```
public void changeToScene (string LoadLevel) {
    Application.LoadLevel (LoadLevel);
}
public void exit() {
    Application.Quit ();
}
```

## Menú Pausa

El menú de pausa es una modificación del script “Pause Menu” disponible en la Asset Store de Unity. Contiene tres opciones: Menú Principal, Salir y Cambiar Calidad Gráfica (opción que posee un submenú en el que elegir la opción gráfica deseada).

El código del script “PauseMenu” se puede ver dividir tres secciones para analizarlo más fácilmente. La primera corresponde a la función que permite activar y desactivar el menú de pausa. La segunda sección sería la encargada de mostrar por pantalla los botones correspondientes a las opciones. Por último, se encuentran las opciones de cambio de calidad gráfica.

**Activar/desactivar menú:** En esta función la única modificación realizada ha sido la selección de la tecla que queremos que active el menú de pausa (ESC), el resto de código no ha sido modificado, ya que su función que si el juego (la visita virtual) está en funcionamiento debe bloquear el movimiento, desactivar el sonido y desactivar el cursor. Si por el contrario el juego está pausado, vuelve a activar el movimiento, sonidos y cursor para seguir el funcionamiento habitual.

```
//check if pause button (escape key) is pressed
if (Input.GetKeyDown("escape")) {

    //check if game is already paused
    if (pauseEnabled == true) {
        //unpause the game
        pauseEnabled = false;
        Time.timeScale = 1;
        AudioListener.volume = 1;
        Screen.showCursor = false;
    }

    //else if game isn't paused, then pause it
    else if (pauseEnabled == false) {
        pauseEnabled = true;
        AudioListener.volume = 0;
        Time.timeScale = 0;
        Screen.showCursor = true;
    }
}
```

**Opciones de menú:** Las modificaciones realizadas en esta parte del han sido la asignación del nombre de los botones, el tamaño que deben ocupar los botones en pantalla y la acción que deben realizar cuando se pulsa sobre ellos.

Cuando se ha comprobado que la pausa está activa, se muestran los rectángulos que contendrán el texto de cada opción mostrada (el título del menú “Pausa”, “Menú Principal”, “Cambiar Calidad Gráfica” y “Salir”) según la configuración de tamaño de cada caja. Además, en cada opción, está configurada la acción a realizar cuando se hace clic sobre ellos: Menú Principal carga Application.LoadLevel(“Menu”) que nos lleva al menú inicial; Salir ejecuta Application.Quit para cerrar la visita virtual por completo; y Cambiar Calidad Gráfica despliega el submenú con las opciones gráficas o lo repliega si se vuelve a hacer clic sobre él. En el código mostrado a continuación existe una línea llena de asteriscos que indica que en ese espacio se encuentra el código referente al menú desplegable que se explicará posteriormente.

```

        if(pauseEnabled == true){

            //Make a background box
            GUI.Box(Rect(Screen.width /2 - 100,Screen.height /2 - 100,250,200),
"Pausa");

            //Make Main Menu button
            if(GUI.Button(Rect(Screen.width /2 - 100,Screen.height /2 -
50,250,50), "Menu Principal")){
                Application.LoadLevel("Menu");
            }

            //Make Change Graphics Quality button
            if(GUI.Button(Rect(Screen.width /2 - 100,Screen.height /2
,250,50), "Cambiar Calidad Grafica")){

                if(showGraphicsDropDown == false){
                    showGraphicsDropDown = true;
                }
                else{
                    showGraphicsDropDown = false;
                }
            }

            *****

            //Make quit game button
            if (GUI.Button (Rect (Screen.width /2 - 100,Screen.height /2 +
50,250,50), "Salir")){
                Application.Quit();
            }
        }
    }

```

**Submenú Cambiar Calidad Gráfica:** En esta sección del script las modificaciones han sido mínimas, reduciéndose a configurar el tamaño de los botones y el texto que debe mostrar cada uno, ya que las rutinas que se ejecutan hacen referencia a configuraciones predefinidas por Unity. El script se encarga también de gestionar la apertura y cierre del submenú de las opciones gráficas, así como la llamada a la configuración de las distintas configuraciones gráficas disponibles, que van a ser explicadas después del código.

```

        if(showGraphicsDropDown == true){
            if(GUI.Button(Rect(Screen.width /2 + 150,Screen.height /2
,250,50), "Muy Baja")){
                QualitySettings.currentLevel = QualityLevel.Fastest;
            }
            if(GUI.Button(Rect(Screen.width /2 + 150,Screen.height /2 +
50,250,50), "Baja")){
                QualitySettings.currentLevel = QualityLevel.Fast;
            }
            if(GUI.Button(Rect(Screen.width /2 + 150,Screen.height /2 +
100,250,50), "Normal")){
                QualitySettings.currentLevel = QualityLevel.Simple;
            }
            if(GUI.Button(Rect(Screen.width /2 + 150,Screen.height /2 +
150,250,50), "Alta")){
                QualitySettings.currentLevel = QualityLevel.Good;
            }
        }
    }

```

```

        if (GUI.Button(Rect(Screen.width / 2 + 150, Screen.height / 2 +
200, 250, 50), "Muy Alta")) {
            QualitySettings.currentLevel = QualityLevel.Beautiful;
        }
        if (GUI.Button(Rect(Screen.width / 2 + 150, Screen.height / 2 +
250, 250, 50), "Excelente")) {
            QualitySettings.currentLevel = QualityLevel.Fantastic;
        }

        if (Input.GetKeyDown("escape")) {
            showGraphicsDropDown = false;
        }
    }
}

```

En esta parte final se va a explicar las características de cada opción gráfica, para ayudar a entender las diferencias entre cada una:

**Muy Baja:** Sin sombras. Sólo iluminación ambiental. Calidad de las texturas baja. Se realiza el cambio ejecutando QualityLevel.Fastest.

**Baja:** Sin sombras. Sólo iluminación ambiental. Calidad de las texturas alta. Se realiza el cambio ejecutando QualityLevel.Fast.

**Normal:** Sombras de baja calidad. Iluminación completa. Calidad de las texturas alta. Distancia baja de renderizado. Se realiza el cambio ejecutando QualityLevel.Simple.

**Alta:** Es la calidad gráfica por defecto con la que se carga la visita. Sombras de calidad media. Iluminación completa. Calidad de las texturas alta. Distancia media de renderizado. Se realiza el cambio ejecutando QualityLevel.Good.

**Muy alta:** Sombras de alta calidad. Iluminación completa. Calidad de las texturas alta. Distancia alta de renderizado. Se realiza el cambio ejecutando QualityLevel.Beautiful.

**Excelente:** Sombras de muy alta calidad. Iluminación completa. Calidad de las texturas alta. Distancia completa de renderizado. Se realiza el cambio ejecutando QualityLevel.Fantastic.



# Anexo VI: Manual de usuario

En este anexo se explicarán brevemente los pasos necesarios para poder utilizar la aplicación correctamente.

## *Inicialización:*

Para acceder a la “Visita Virtual” sólo es necesario hacer doble clic sobre el ejecutable “VisitaVirtual.exe” para que la aplicación se inicie. No requiere ninguna instalación.

## *Menús:*

### *Menú principal:*

Está disponible nada más acceder a la aplicación, y las opciones disponibles en este menú son:

- **Visita Virtual:** Acceder a la visita para poder explorar el edificio Betancourt.
- **Salir:** Cerrar la aplicación.

### *Menú pausa:*

Para acceder a este menú es necesario pulsar la “tecla ESC” mientras se está realizando la visita virtual. De igual manera, pulsado ESC se vuelve a la visita. Las opciones disponibles son:

- **Menú Principal:** Redirige la visita al menú principal.
- **Cambiar Calidad Gráfica:** Permite modificar la calidad gráfica de la visita para adecuar el rendimiento de la aplicación a las características técnicas del equipo.
- **Salir:** Cerrar la aplicación.

## *Interacción:*

- **Mover personaje:**
  - Avanzar: Tecla W y flecha arriba ↑
  - Retroceder: Tecla S y flecha abajo ↓
  - Desplazamiento izquierda: Tecla A y flecha izquierda ←
  - Desplazamiento derecha: Tecla D y flecha derecha →
- **Mover cámara:** Movimiento de ratón.
- **Interacción:**
  - Abrir/cerrar puertas dobles: Tecla E.
  - Abrir/cerrar puertas automáticas: Apertura y cierre automático por proximidad.